Evolutionary Synthesis of HVAC System Configurations: Algorithm Development (RP-1049)

Jonathan Wright, PhD, CEng   **Yi Zhang, PhD**   **Plamen Angelov, PhD**
Member ASHRAE   Member ASHRAE

Victor Hanby, PhD, CEng   **Richard Buswell, PhD**

This paper describes the development of a model-based optimization procedure for the synthesis of novel heating, ventilating, and air-conditioning system configurations. The optimization problem can be considered as having three suboptimization problems: the choice of a component set; the design of the topological connections between the components; and the design of a system operating strategy. In an attempt to limit the computational effort required to obtain a design solution, the approach adopted in this research is to solve all three subproblems simultaneously. The computational effort has been further limited by implementing simplified component models and including the system performance evaluation as part of the optimization problem (there being no need, in this respect, to simulate the system performance).

The optimization problem has been solved using a Genetic Algorithm (GA) that has data structures and search operators specifically developed for the solution of HVAC system optimization problems. The performance of the algorithm and various search operators has been examined for a two-zone optimization problem, the objective of the optimization being to find a system design that minimizes system energy use. In particular, the performance of the algorithm in finding feasible system designs has been examined. It was concluded that the search was unreliable when the component set was optimized, but if the component set was fixed as a boundary condition on the search, then the algorithm had an 81% probability of finding a feasible system design. The optimality of the solutions is not examined in this paper but is described in an associated publication (Wright and Zhang 2008). It was concluded that, given a candidate set of system components, the algorithm described here provides an effective tool for exploring the design of novel HVAC systems.

INTRODUCTION

The capital cost, energy use, or life-cycle cost of a building can be minimized through the use of simulation-based numerical optimization in which a simulation is used to evaluate the performance of the building for each trial solution of the optimization. To date, the approach has been applied to (1) the optimization of the building fabric construction (Caldas and Norford 2002, 2003; Wetter and Wright 2004; Wang et al. 2005); (2) the sizing of heating, ventilating, and

Jonathan Wright is a professor of building optimization and **Richard Buswell** is a lecturer for the Department of Civil and Building Engineering, Loughborough University, United Kingdom. **Yi Zhang** is a senior research fellow and **Victor Hanby** is a professor at the Institute of Energy and Sustainable Development, De Montfort University, Leicester, United Kingdom. **Plamen Angelov** is a senior lecturer in the Department of Communications Systems at Lancaster University, United Kingdom.

air-conditioning (HVAC) systems (Wright and Hanby 1987; Wright 1996); (3) HVAC system supervisory control (Nassif et al. 2005); (4) the simultaneous optimization of building construction, HVAC-system size, and system supervisory control (Wright and Farmani 2001; Wright et al. 2002); and the optimization of building layout (Jo and Gero 1998; Gero and Kazakov 1998). However, to date, no research exists for the synthesis of optimum HVAC system designs.

A review of existing HVAC system designs (Wright et al. 2004) indicates that a variety of HVAC configurations have evolved to match the diversity in building design and the thermal load conditions. However, there is no guarantee that for every new building an optimum HVAC system configuration already exists. Optimum system performance can only be guaranteed if the HVAC system is synthesized to capitalize on the thermal characteristics of the building. This paper describes an approach to the synthesis of optimum HVAC system configurations for minimum energy use (the application of the approach to the synthesis of an example HVAC system is described in Wright and Zhang (2008).

Development of the optimization procedure has been informed by the characteristics of the design problem and a perceived need to limit the computational effort associated with simulating the system performance (any optimization will require the performance simulation of many alternative system designs).

OPTIMIZATION APPROACH

In general, there are three elements to the synthesis of a new HVAC system as follows:

1. Selection of appropriate system components (heating and cooling coils, humidifiers, air mixing sections, etc.)
2. Definition of the topological connections between the components
3. Optimization of the full load capacity of the components and the system control strategy

Since a topology cannot be formed until the component set has been selected, and the system capacity and operation cannot be optimized until the topology has been defined, the design of such systems is usually performed sequentially. The sequential design process is generally reflected in an optimization procedure that iterates among the three design tasks (e.g., as in the approach described by Diwekar et al. [1992]).

Figure 1 illustrates the iterative optimization approach in which there are three nested optimizations, each optimization being represented by a separate *Do{…}Until* loop. The highest-level loop optimizes the choice of component set; the second level optimizes the system topology; and the lowest level optimizes the system capacity and operation. The *Vary(.)* functions represent the operators of the particular optimization algorithm that generate the new solutions (it being possible to use different optimization algorithms for each subproblem). Note that the system capacity and operation cannot be optimized until a feasible topology has been found (a feasible topology being one that provides a workable system configuration). The system control and capacity optimization is also constrained in that the system must have sufficient capacity and be operated in a manner that maintains the zone thermal comfort (the system components must also operate within their performance limits). Also, note that the optimality of each optimization process is a function of the system performance as evaluated by the performance simulation in the system capacity and control optimization loop.

The iterative optimization procedure in Figure 1 has a high computational overhead in that the sub-optimization problems must be repeated for each trial solution of the upper loop(s). In particular, there are likely to be many alternative topologies for a given component set with the result that the topology optimization may require many trial solutions and associated suboptimizations of the system capacity and control strategy. Therefore, in order to reduce the number of
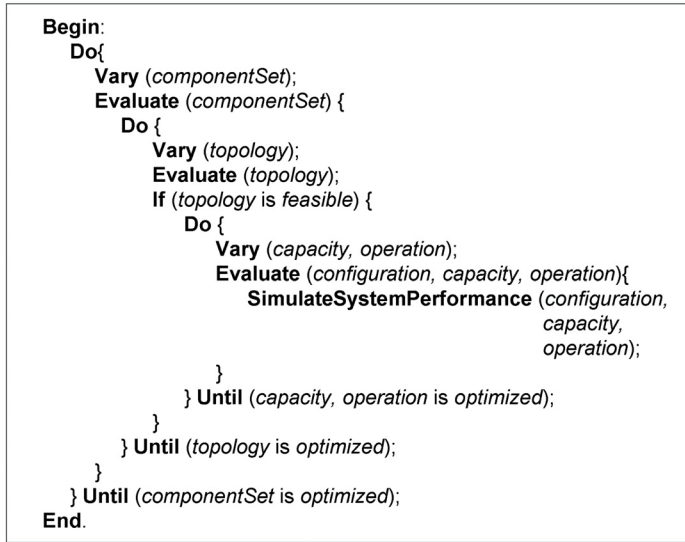
```
Begin:
   Do{
       Vary (componentSet);
       Evaluate (componentSet) {
          Do {
              Vary (topology);
              Evaluate (topology);
              If (topology is feasible) {
                  Do {
                      Vary (capacity, operation);
                      Evaluate (configuration, capacity, operation){
                          SimulateSystemPerformance (configuration,
                                                     capacity,
                                                     operation);
                      }
                  } Until (capacity, operation is optimized);
              }
          } Until (topology is optimized);
       }
   } Until (componentSet is optimized);
   End.
```

Figure 1. Iterative optimization process.

system performance simulations that would result from the repeated optimization of the system capacity and control strategy, in this research the component set, system topology, capacity, and control strategy have been optimized simultaneously. Figure 2 illustrates the general form of the algorithm, there being only one optimization—a *Do{…}Until* loop. Although this procedure should result in fewer system performance simulations than the iterative approach, it has the disadvantage that the optimization is more constrained than any one suboptimization in the iterative approach and, therefore, is potentially more difficult to solve.

SYSTEM MODELING

The simplification of the system models and simulation procedure is necessary to reduce the computational time spent in performing the many system simulations. The modeling approach adopted here is also a function of the manner in which the system-control variables and strategy are defined.

Representation of an Optimum Control Strategy

Since the majority of system configurations synthesized by the optimization will be novel, it may not be possible to implement a conventional control strategy for every new configuration. Therefore, in this research, the controlled variables for each load condition are optimized directly by the optimization algorithm (the optimization algorithm therefore acts as an optimum controller). In this study, the control variables are the capacity of the components and the air-mass flow rates through each part of the system. The practicable implementation of the optimized control strategy would have to be inferred *a posteriori* to the optimization. It may not be possible to replicate exactly the optimized control strategy using contemporary control systems. However, the approach adopted here ensures that the optimum performance and, therefore, full potential of each system is used in evaluating each trial system during the optimization (any difference in performance of the optimized systems should then be due to the system configuration and not the manner in which it is controlled).
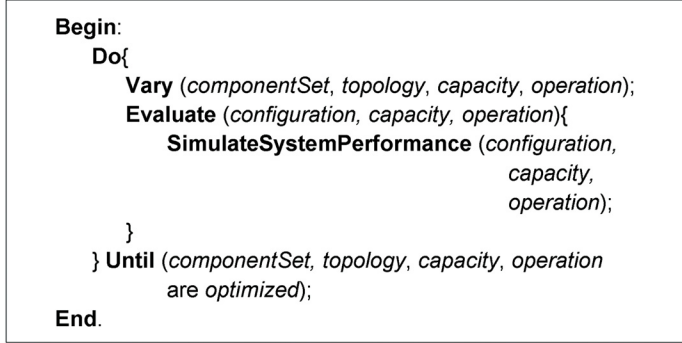
```
Begin:
   Do{
      Vary (componentSet, topology, capacity, operation);
      Evaluate (configuration, capacity, operation){
         SimulateSystemPerformance (configuration,
                                             capacity,
                                             operation);
      }
   } Until (componentSet, topology, capacity, operation
          are optimized);
End.
```

Figure 2. Simultaneous optimization process.

System Performance Simulation

Although the simultaneous optimization procedure (Figure 2) limits the number of system simulations required by the optimization, the system simulation remains the computationally dominant element of the optimization. The computationally intensive part of a system simulation is associated with the iterative procedure used to solve the system performance equations. In this research, the computational overload associated with the system performance simulation has been limited by the following:

1. Using simplified steady-state process models for each system component, a simplified model having fewer equations to solve than a detailed model.
2. Solving the system performance equations simultaneously as part of the optimization problem and without the need for iteration between the equations.

The simplified process models (Zhang 2005) have three elements: the thermal capacity of the component, an input-output psychometric process model to calculate the condition of the air leaving the component, and an air-mass flow rate balance and air-pressure loss model. The controlled variables associated with the simplified models are the thermal capacity of the components and air-mass flow rates at a particular load condition. Note that the term *capacity* is used here to indicate the capacity at a given load condition rather than just at the peak load condition.

The need to iterate on the system performance equations is associated with a cycle in the system, the only source of cycles resulting from the simplified models being the recirculation of air from the zone back through the system. Therefore, iteration between the system equations can be avoided by breaking the air cycle(s) at the entry to the zone(s) and assuming that the air leaving the zone is at the desired temperature and humidity. Given that the component capacity and air-mass flow rates have been set by the optimization (since they are the controlled variables), the component input-output process models can then be used sequentially to determine the supply air condition. However, it may be the case that, at a particular point in the optimization, the specified component capacities and air-mass flow rates result in the supply-air condition that cannot meet the zone loads. The optimization must therefore be constrained to ensure that the supply-air condition meets the zone loads.

For example, Figure 3 illustrates a novel single-zone system consisting of a heating coil, two cooling coils, two pair of flow-mixing and flow-diverting duct branches, and the zone (the supply and extract fans are not illustrated). The component capacities $(\dot{Q}_1, \dot{Q}_2, \dot{Q}_3)$, and air-mass
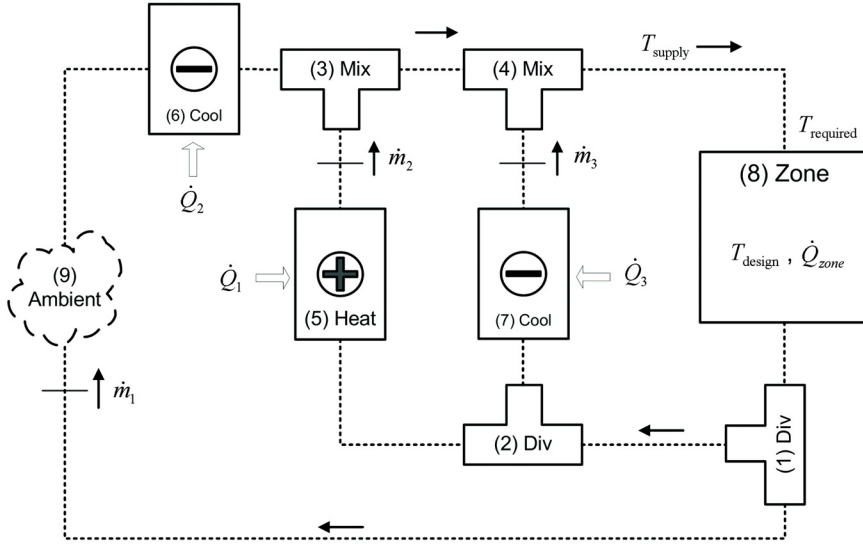
Figure 3. Novel single-zone system.

flow rates $(\dot{m}_1, \dot{m}_2, \dot{m}_3)$ are set by the optimization, with the choice of which air mass flow rates to optimize having been decided by a graph-theoretic analysis of the system topology. Given the ambient temperature ($T_{ambient}$), the zone design temperature (T_{design}), the component capacities, and the air-mass flow rates, it is possible to calculate the supply-air condition (T_{supply}). A comparison of this to the supply-air temperature necessary ($T_{required}$) to offset the zone load ($\dot{Q}_{zone}$) can be used to form a constraint on the optimization so that $T_{supply} \approx T_{required}$. Note that, although this example is for the sensible heat only, the latent heat is also included in the component models.

Fan-Duct System Model

Given that we seek to simplify the system simulation, the fan and associated duct flow-pressure relationships have not been modeled explicitly. However, the fan energy use is calculated as a function of the pressure drop across each system component, the pressure drop being a function of the air-mass flow rate through the component.

SELECTION OF AN OPTIMIZATION ALGORITHM

The choice of an optimization algorithm is in general dependent on the characteristics of the optimization problem (in terms of the nature of the problem variables, the problem constraints, and the problem objective function).

There are three groups of problem variables associated with the optimization: the component set; the system topology; and the component capacities and system air-mass flow rates (there being separate capacity and flow-rate variables for each load condition). The component set can be represented by a simple discrete index variable that points to the component type; the topology can also be represented by discrete index variables in the form of an adjacency matrix. However, the component capacities and air-mass flow rates are continuous variables and, therefore, the optimization problem is mixed-integer. The dimensionality of the problem is also high, with some problems considered in this research having in the order of 1019 alternative topologies (Wright et al. 2004; Wright and Zhang 2008).

The objective function in this research is the system energy use, which is a nonlinear function of the problem variables. The problem is highly constrained, with constraints maintaining the feasibility of the topology and the viability of the supply-air conditions and ensuring that the system components operate within practicable performance limits. The search space is also highly multimodal in that each feasible topology will result in a local minimum for system energy use.

The solution of such mixed-integer, nonlinear, multimodal, high-dimensionality topological optimization problems is almost exclusively limited to the population-based, probabilistic evolutionary algorithms (EA). Perhaps the most successful EA to date for the solution of topological optimization problems is that of genetic programming (GP) (Koza 1992). GP is a form of EA in which the solutions are represented as parse trees. GP has been applied successfully to a large number of difficult topological optimization problems (Koza et al. 2003). However, the approach is highly computationally intensive and produces output that can be difficult to interpret. Therefore, the approach adopted in this research is to develop a problem-specific EA that is more computationally effective than a generic GP algorithm. The selected algorithm is based on an EA form closely related to GP—a genetic algorithm (GA).

Genetic Algorithms

A detailed introduction to GAs can be found in many texts (among others, Goldberg [1989]; Bäck [1996]; Deb [2001]; and De Jong [2006]). Briefly, GAs operate on a set of solutions (a population) using probabilistic search operators. Figure 4 illustrates the main algorithm operators. First, the population of solutions is initialized (normally by randomly selecting variable values to lie within specified bounds). Second, the solutions are evaluated and a fitness is assigned to each solution, fitness being a relative measure of the optimality of a solution in comparison to the optimality of the other solutions in the population. In the case of this research, fitness assignment requires the system energy use and constraint function values to be calculated. Fitness assignment is followed by a selection procedure in which pairs of solutions are probabilistically selected from the population, where the chance of an individual being selected is in some proportion to its fitness. The pairs of selected solutions are then used to form two new solutions in a process of recombination. However, since the recombination process is probabilistic, it is possible that
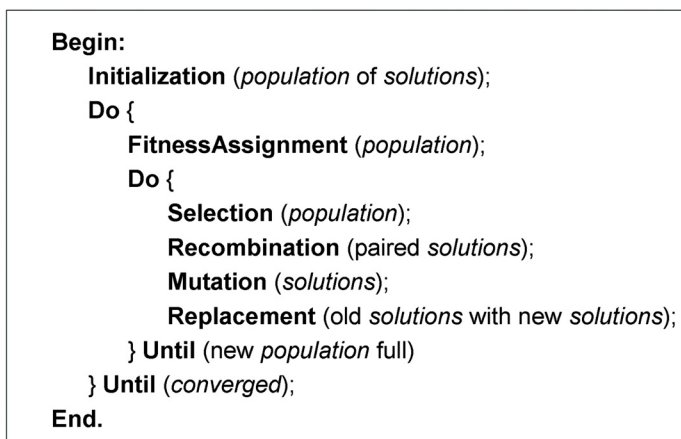
```
Begin:
    Initialization (population of solutions);
    Do {
        FitnessAssignment (population);
        Do {
            Selection (population);
            Recombination (paired solutions);
            Mutation (solutions);
            Replacement (old solutions with new solutions);
        } Until (new population full)
    } Until (converged);
End.
```

Figure 4. A genetic algorithm.

recombination does not take place and that the new solutions are identical to the selected pair of individuals (this also occurs when the selected individuals are identical). The variable values of the recombined solutions are then probabilistically perturbed in a process known as mutation. Finally, two old solutions from the population are replaced with the newly created solutions. The process of selection, recombination, mutation, and replacement continues until the population has been completely replaced with new solutions. Note that, since recombination and mutation are probabilistic, some solutions may be identical to those found in the old population. It is also common for the replacement operators to ensure that the best solution found to date is not replaced and therefore is retained in the population. The algorithm continues to iterate on the population until convergence is reached (each new population is termed a *generation* or *epoch*). Convergence is often arbitrary judged and is defined to occur after the algorithm has completed a specified number of generations.

ALGORITHM DESIGN

A number of alternative forms of GA exist, the differences being in the encoding of the problem variables as well as the form of GA operators. In this research, a problem-specific GA is implemented using a problem-specific encoding of the problem variables together with problem-specific fitness assignment, recombination, and mutation operators.

Problem Variable Encoding

In GAs, the problem variables are encoded in data structures known as chromosomes. Both integer and continuous variables are commonly encoded as a binary string (Goldberg 1989; Bäck 1996; DeJong 2006). However, a binary chromosome datastructure is ineffective in the solution of topological optimizations problems, the most effective representation for a topology being an integer vector (Michalewicz and Fogel 2000). Since the problem space in this research is mixed integer, a multichromosome datastructure has been implemented that consists of two integer vector chromosomes, one each for the component set and system topology, and real-value vector chromosomes for the airflow rate and component capacity variables (each load condition being represented by a separate flow-capacity chromosome). The multichromosome genome has been implemented in a generic and flexible manner (Wright et al. 2004; Zhang 2005), the most important feature of which is that each member chromosome can be associated with its own unique set of recombination and mutation operators. This feature is important, not only in that different search operators are required for integer and real-value chromosomes, but also because separate operators can be developed to specifically exploit the characteristics of each subproblem and improve the robustness and convergence rate of the GA (the subproblems for the component set being topology, flow rates, and capacity).

The structure of the genome is described here by example (a formal definition of the genome structure is given by Wright et al [2004] and Zhang [2005]). Figure 5 gives the genome for the single-zone system illustrated in Figure 3. The component set chromosome is a simple integer vector in which the integer values point to a component type—e.g., a value of 0 points to a flow-diverting duct branch (known here as a *Div* or *diverting T*). Component types 1, 2, and 3 in the example point to the flow-mixing duct sections (a *Mix* or *mixing T*), the heating coil, and the cooling coils. The component set chromosome also contains pointers to the zone(s) and the ambient environment (types 7 and 11, respectively). Strictly, these are not included in the problem variables, as they must always be included in the component set.

The topology chromosome is in the form of a modified adjacency matrix, the chromosome structure being developed to limit the redundancy and duplication of topological connections that are present in a conventional adjacency matrix. Each entry in the chromosome indicates a connection between two components and, in particular, the connection of the outlet of one component to
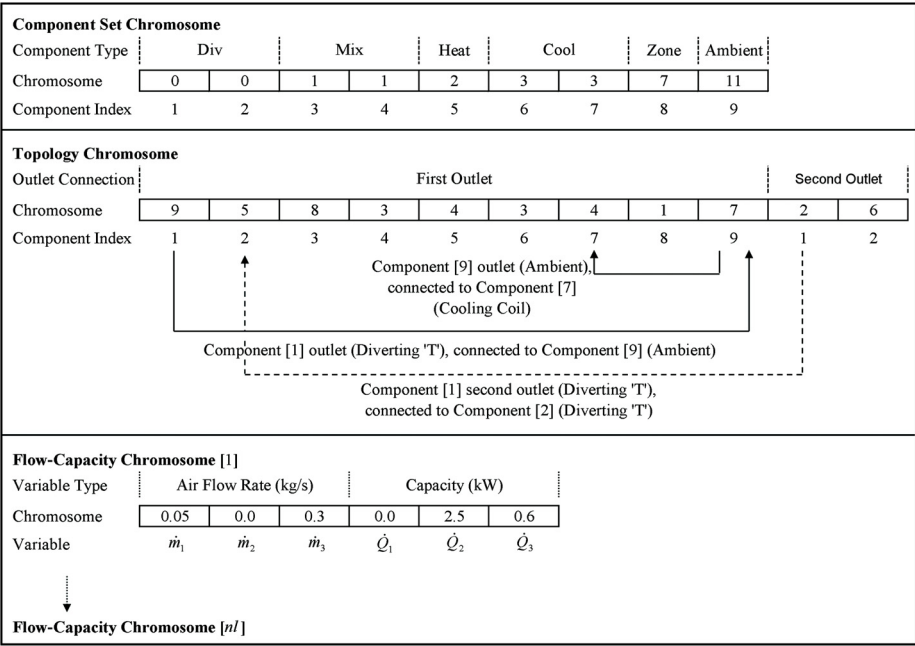
Figure 5. Example genome.

the inlet of another component. The order of entries corresponds to the order of components given in the component set chromosome. Since the example system has nine components, including the zone and the ambient environment, the components are indexed in order of one to nine. However, the topology chromosome is typically longer than the number of components, since the flow-diverting duct branches have two outlets that must be connected to components (the connection for the second outlet being defined after the first set of connections).

The topology chromosome can be interpreted by reference to the component set chromosome. For example, the first entry in the topology chromosome has a value of nine, this indicates that the first outlet of Component 1 is connected to the inlet of Component 9. The component set chromosome indicates that Component 1 is of Type 0 (a diverting T) and that Component 9 is of Type 11 (ambient); hence, the first outlet of the diverting T is connected to the ambient (as in Figure 3). The outlet of the ambient environment is in turn connected to Component 7, which, from the component set chromosome, is of Type 3, a cooling coil. The connection of the second outlet of Component 1 (diverting T), is specified by the tenth entry in the topology chromosome. This indicates that the second outlet is connected to Component 2, which, from the component set, is also of Type 0, another diverting T.

The airflow rate and component capacity control variables are encoded in nl real value chromosomes, with each chromosome representing the control values for a single load condition (nl being the number of load conditions). The flow-capacity chromosomes are in two parts, the first section being for the air-mass flow rates and the second part being for the component capacities. The component capacities are given in the order of heating coils, cooling coils, and steam humidifiers. The example system has one heating coil, two cooling coils, and no humidifiers (Figure 3), giving three capacities to be controlled at each load condition. The three capacities in the example flow-capacity chromosome (Figure 5), therefore, correspond to the heating coil

(component index 5 of the component set chromosome) and the two cooling coils (having corresponding component set indexes of 6 and 7).

Although we seek to assign all air-mass flow rates in the system, it is only necessary to control a subset of the flow rates; the remaining flow rates are determined from the flow-mixing or flow-diverting process models. The number of flow rates that must be set is equal to $n_{mix} + 1$, where n_{mix} is the number of air-mixing or air-diverting components. The example genome (Figure 5) indicates that, for the example system (Figure 3), there are three airflow rate control variables (there being two air-mixing sections). The first of the air-mass flow rates of the chromosome is always the ambient airflow rate at inlet to the system[1]. The choice of the remaining two flow rates and their corresponding duct sections is determined through a graph-theoretic tree-growing algorithm (Wright et al. 2004; Zhang 2005).

Problem Variable Definition. There are three groups of problem variables encoded in three types of chromosome: the component set, the system topology, and the airflow rate and component capacity control variables. The problem variables can be formally defined as follows:

$$X = (X_{set}, X_{top}, X_{con}) \tag{1}$$

where X_{set}, X_{top}, and X_{con} are the component set, topology, and control variables, respectively, such that

$$X_{set} = (x_{set,\,1}, \cdots, x_{set,\,n_{set}}) \in Z^{n_{set}}$$

$$X_{top} = (x_{top,\,1}, \cdots, x_{top,\,n_{set}}) \in Z^{n_{top}} \,, \tag{2}$$

$$X_{con} = (x_{con,\,1}, \cdots, x_{con,\,n_{set}}) \in \Re^{n_{con}}$$

where

n_{set} = the number of component set variables,
n_{top} = the number of topology variables, and
n_{con} = the number of control variables.

The number of topology variables is a function of the number of components; number of air mixing components, n_{comp}; and number of air mixing components, n_{mix}. The number of control variables is a function of the number of heating coils, n_{heat}; the number of cooling coils, n_{cool}; the number of humidifiers, n_{humid}; the number of air-mixing sections, n_{mix}; and the number of load conditions, n_{load}, as follows:

$$n_{top} = n_{comp} + n_{mix},$$
$$n_{con} = n_{load} \times (n_{heat} + n_{cool} + n_{humid} + n_{mix} + 1) \tag{3}$$

Fitness Assignment

The fitness of a solution is a measure of its optimality as compared to the other solutions in the population. Since the optimization problem considered here is a constrained optimization problem, the fitness of a solution is a function of the constraint values as well as the objective function value.

---

1. In this research, it is assumed that the exhaust airflow rate equals the inlet airflow rate.

Objective Function Formulation. In this research, the objective function, $f(x)$, the value of which is to be minimized, is the system annual energy consumption. Since a separate flow-capacity chromosome is required for each load condition, it is impracticable to optimize the system operation for every hour of the year (as the resulting number of problem variables would make the problem difficult to solve). Therefore, in this research, the annual energy use is approximated using a weighted sum of the component capacities over a limited number of load conditions as follows:

$$\min f(X) = \sum_{i=1}^{n_{load}} w_i \left[\sum_{j=1}^{n_{h+c+h}} \dot{Q}_{i,j} + W_{fan,i} \right] \tag{4}$$

where

w_i	=	the weight for load condition i
$Q_{i,j}$	=	the capacity of component j at load condition i
$W_{fan,i}$	=	the total fan power at load condition i
$n_{h+c+h} = n_{heat} + n_{cool} + n_{humid}$	=	(the sum of the number of heating coils, cooling coils, and humidifiers)
n_{load}	=	the number of load conditions

The total fan power is calculated as a function of the pressure loss and airflow rates through each component and a notional length of ductwork.

The weights are formulated such that $\sum_{i=1}^{n_{load}} w_i = 1.0$ and, as such, the objective function is a weighted average of the capacities over the load conditions. In the example optimization problems studied in this research (Wright and Zhang 2008), the annual energy performance is approximated from nine load conditions. The nine loads conditions approximate to morning, early afternoon, and early evening operation in each of three seasons, winter, summer, and a transition season.

Constraint and Infeasibility Functions. As well as each problem variable being constrained in a specified range of values, there are three sets of problem constraints as follows:

1. Constraints on the viability of the system topology
2. Constraints on the procedure for evaluating the system performance
3. Constraints on the system operation

The topology constraints are concerned with ensuring that system components are connected in a manner that enables feasible operation. The topology is defined by a directed graph, which must be strongly connected to ensure that air can be circulated between any two of the components in the system (the ambient environment and zone(s) being considered as components in the system). The occurrence of ineffective structures in the system is also constrained; these structures are associated with the circulation of air in the duct system that do not result in a change in the air condition—e.g., an airflow being split in a duct branch only to be merged again immediately downstream. Each of the topology constraints is represented as an equality constraint (a binary constraint with a value [0, 1]). The topology constraint violations are aggregated to give a single normalized value of the violation. The aggregated value is in the range 0.0 to 1.0 with a value of zero indicating a feasible topology.

The system performance evaluation constraints are concerned with identifying failure in the evaluation of system performance. This can occur for overly complex systems or when the procedure for assigning airflow rates to particular branches results in insolvable airflows. The

performance evaluation constraints are also aggregated and normalized to be in the range 0.0 to 1.0 (0.0 indicating complete feasibility).

The system operating constraints are concerned with ensuring that all components operate within their performance limits and that the system is able to maintain the desired zone conditions (the zone constraints being on the amount of outside air entering the zone and the temperature and humidity of the supply air entering each zone). The operation constraints are in the form of both inequality and equality constraints, although a reasonable engineering tolerance has been applied to the equality constraints. The system performance and operation constraints have also been aggregated to have a value in the range 0.0 to 1.0 (0.0 indicating complete feasibility).

Note that, although there are three groups of constraints (topology constraints, performance evaluation constraints, system operation constraints), the system performance can only be evaluated if a feasible topology exists. Similarly, the value of the system operating constraints can only be calculated if the system performance evaluation has been successfully completed. The discontinuity of the constraint space is reflected in the formulation of the infeasibility function. If $c_{top}(X_{set}, X_{top})$, $c_{ev}(X)$ and $c_{op}(X)$ are the aggregated constraint values for the topology, system performance evaluation, and system operating constraints, respectively, then the system infeasibility $i(X)$, is given as follows:

$$i(X) \; = \; \begin{cases} 0.9 + 0.1c_{top}(X_{set}, X_{top}), & \text{if}(c_{top}(X_{set}, X_{top}) > 0.0) \\ 0.45 + 0.45c_{ev}(X), & \text{else if}(c_{ev}(X) > 0.0) \\ 0.45c_{op}(X), & \text{else} \end{cases} \tag{5}$$

The infeasibility function in Equation 5 gives three bands of infeasibility: infeasible topologies result in an infeasibility value in the range 0.9 to 1.0, failure to evaluate the system performance results in a value in the range 0.45 to 0.9, and infeasible system operation results in a value in the range >0.0 to 0.45. Note that the choice of banding should not impact the performance of the search when the fitness is based on a rank order of the solutions, as in the case of this research.

Stochastic Ranking. The fitness of a solution in the population is a function of its objective function and infeasibility values compared to the values for all other solutions in the population. The fitness assignment adopted here is based on the stochastic ranking procedure (Runarsson and Yao 2000), one of the most robust fitness assignment algorithms for constrained optimization. The procedure rank orders the solutions with, for this minimization problem, the lowest ranked solution being the fittest. The ranking is achieved by a probabilistic bubble sort, with the comparison of two solutions during the sort being made on either the objective function values or the infeasibility values. If both solutions are feasible, then the comparison is always made on the objective function values. However, if one or more of the solutions in the comparison is infeasible, then the choice as to whether the two solution are compared on their objective or infeasibility values is decided probabilistically (the probability of the comparison being made on the degree of infeasibility being a parameter of the method). This ensures that, during the early stages of the search, the search direction is not overly biased by the need to find a feasible solution and that, for example, slightly infeasible low objective function value solutions have some chance of influencing the search direction.

Fitness "Ageing." During development of the GA, it was observed that, although the synthesized systems had an energy consumption comparable to established system configurations, the synthesized systems were suboptimal. An analysis of the algorithm behavior with respect to the exploration of different topologies indicated that the search could become dominated by a single topology from an early stage of the search. This led to the introduction of a

new algorithm operator that penalizes the fitness of solutions that have the dominant topology. The penalty or *ageing* function is given by the following:

$$j'(.) = j(.) \times max[(n_e - q \times n_g), 1.0] \tag{6}$$

where

$j(.)$ =   the fitness of a solution
$j'(.)$ =   the solution's fitness after ageing
n_e =   the number of times the solution's topology has been evaluated
q =   the maximum number of evaluations per generation allowed for a given topology
n_g =   the number of generations to date

The effect of the function is that, provided the number of evaluations to date (n_e) is less than the maximum allowed for the current generation ($q \times n_g$), the fitness remains unchanged. If, however, the number of evaluations exceeds the maximum allowed, then the fitness function value is increased by a factor equal to the number of evaluations in excess of the limit ($n_e - q \times n_g$). Note that, in this optimization problem, the fitness is minimized since we seek to minimize the objective function value. Implementation of the ageing operator led to an order of magnitude increase in the number of topologies explored during the search and, as a result, a significant improvement in the optimality of the synthesized systems (Wright and Zhang 2005).

Selection

Solutions are selected for recombination and mutation using the established tournament selection operator (Bäck 1996; DeJong 2006). Tournament selection operates by randomly sampling *n* solutions from the population, the winner of the tournament and solution used in recombination being the solution with the highest fitness (the other *n*-1 solutions are rejected).

Recombination

Recombination is a process by which problem variable values from two chromosomes are used to produce one or more new chromosomes. In this research, the problem variables have been encoded in three distinct groups of chromosome (the component set, the system topology, and the control variables). Not only does this enable the handling of both discrete and continuous problem variables, but also allows separate problem-specific search operators to be applied to each type of chromosome. In the case of the topology and flow-capacity chromosomes, several different recombination operators have been developed and grouped into a hyperoperator in which the choice of operator applied during a particular recombination operation is decided probabilistically (the probability associated with a particular operator has been determined here by small-scale trials of the algorithm).

Component Set Chromosome Recombination. The simplest form of recombination for an integer chromosome is to randomly select a fixed number of problem variables and swap their values in each of the two chromosomes to produce two new chromosomes (this is known as N-point crossover). However, in this research, the number of components and, therefore, chromosome length can vary between solutions in the population. Since the length of two selected chromosomes can be different, simply swapping the variable values is ineffective, as part of the longer chromosome would remain unchanged. The effectiveness of this simple recombination is also limited by the fact that the order of components is fixed in each chromosome and, hence, an N-point recombination may result in the same component types being swapped with no change to the chromosome. In this respect, a new recombination approach has been implemented in which the components found in each chromosome are pooled and then redistributed to create two new chromosomes (Wright et al. 2004; Zhang 2005).

System Topology Chromosome Recombination. The topology chromosome has been implemented to ensure that the topology conforms to the valence rules for a HVAC system, specifically that each component outlet is connected to the inlet of one other component. Although the valence restriction reduces the size of the topology search space, this remains large and is the order of the following (Zhang 2005):

$$O\left(\frac{(n_{comp} + n_{mix})!}{2^{n_{mix}}}\right) \tag{7}$$

where n_{comp} is the total number of components and n_{mix} the number of flow-mixing (or diverting) components. For example, even the simple nine-component, two-flow-mixing section system in Figure 3 has a total search space of 9.97×10^6 solutions. Further, although the form of topology chromosome satisfies the system valence rules and in effect reduces the search space, there is no guarantee that a randomly generated chromosome will satisfy all of the topology constraints. An experimental investigation indicated 99% of randomly generated chromosomes will fail to satisfy one or more of the topology constraints (Zhang 2005). Given this, there is a need to implement recombination operators that tend to preserve the feasibility of a topology. Two recombination operators have been implemented in this research: partially matched crossover and adjacent component crossover. Partially matched crossover is an established operator that is effective in creating new topologies but can result in an unconnected system. Adjacent component crossover addresses this problem in that it tends to create new topologies that remain connected but, as a result, the exploratory power associated with this operator is reduced. In this respect, both operators have been implemented as part of the hyper-recombination with a 0.1 probability that partially matched crossover will take place, compared to a 0.9 probability for the adjacent component crossover (the result being that, in general, the topology recombination preserves the system connectivity but that recombination maintains a degree of its exploratory power).

Partially matched crossover is described in standard texts (Goldberg 1989), whereas adjacent component crossover is described here in relation to the topology given in Figure 6 (which illustrates two similarly connected systems, A and B). The number associated with each component in Figure 6 indicates the component index in the component set. Adjacent component crossover operates by selecting a feature that is present in both topologies (in this case, the feature is a flow-mixing section). The components next to the selected feature are then compared and, if they are different but components of each type exist somewhere in each topology, then they are swapped to produce two new topologies (in this case, A' and B').

Airflow Rate and Component Capacity Chromosome Recombination. Five methods of recombination have been implemented for the airflow rate and component capacity chromosomes, four of which are established methods a the fifth a problem-specific method. The four established methods are the center-of-gravity crossover, blend crossover, arithmetic crossover, and N-point crossover (N-point). Blend crossover, arithmetic crossover, and N-point operators are described in Deb (2001), and the center-of-gravity crossover is described by Angelov (2001). As for the case of the N-point topology crossover, the N-point flow-capacity crossover creates new chromosomes by simply swapping the variable values of two selected (parent) chromosomes. However, blend crossover, arithmetic crossover, and center-of-gravity crossover, all generate new variable values that are based on the interval between the two parent variable values. In the case of arithmetic crossover and center-of-gravity crossover, the new variable values lie within the range of the two parent values, whereas they can lie outside of this range for blend crossover. All four recombination operators are included in a hyper-recombination operator, with the choice of which form of recombination is executed being determined probabilistically

(the probabilities of use for each operator have been set to 0.25 for center-of-gravity crossover, 0.6 for arithmetic crossover, 0.1 for blend crossover, and 0.05 for N-point).

In addition to the four conventional recombination operators, a problem-specific, selective chromosome crossover has been developed and implemented. Rather than operate at the level of individual variables, selective chromosome crossover operates on the whole chromosome. Each operating condition is associated with a separate flow-capacity chromosome, separate energy use (the objective function), and degree of feasible operation (as represented by Equation 5 and, in particular, $c_{op}(X)$). Where the two selected (parent) chromosomes have an identical system configuration[2], then the flow-capacity chromosomes for each load condition can be compared directly and used to form a new genome consisting of the best flow-capacity chromosomes found in either of the two parents. The best chromosome at a given load condition is determined by comparing the infeasibility and objective function values, with the choice of the best chromosome being made on infeasibility when both solutions are infeasible or on the objective function value when both solutions are feasible; if only one of the pair is feasible, it is taken as the best solution. The result of this operator is that the new solution should have better overall performance than either of the two parent solutions.

Mutation

Solutions resulting from the recombination process are mutated in order to maintain the exploratory power of the search (mutation in effect being a disruptive operator that has the potential to change the search direction). As for the recombination operators, several mutation operators, including problem-specific operators, have been implemented in a hypermutation operation for each type of chromosome. The probability of a particular mutation operator being used within the hyperoperator has been determined by small-scale trials of the algorithm.
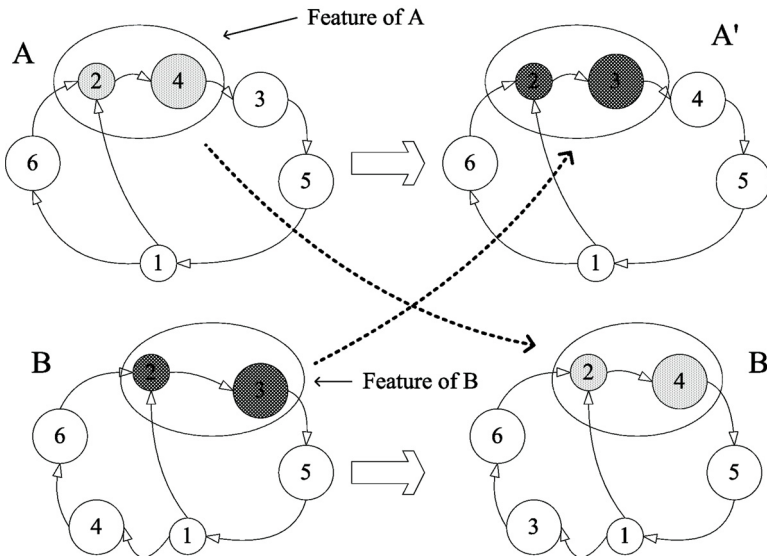

Figure 6. Example of adjacent component crossover.

---

2. A configuration consists of a component set and system topology (that is, the choice of components and their connections).

Component Set Chromosome Mutation. The most commonly used form of mutation operates by probabilistically selecting a problem variable from a chromosome, the value of which is then changed by a random amount. This simple approach has been modified in this research to ensure that the number of each component type found in the system can be controlled and that practicable restrictions can be placed on the types of component included in the system.

System Topology Chromosome Mutation. Three topology chromosome operators have been implemented: a random initialization mutation, a link swap mutation, and a component swap mutation. Random-initialization mutation randomly reinitializes the topology chromosome and, in effect, seeds the search with a completely new topology. Link-swap mutation swaps two of the component outlet links, the outlet links being defined by integer problem variables in the topology chromosome. For example, Figure 7 illustrates a six-component system that includes one flow-mixing section and gives a topology chromosome length of seven problem variables. The number 5 for the third entry in the original chromosome indicates that Component 3 is linked to Component 5. The last entry in the topology chromosome is for the connection of the second outlet of the mixing section (Component 1), which, in this case, is connected to Component 6. In the example given in Figure 7, swapping the links results in a new topology but one that is poorly connected, as there is no flow path from Components 2, 3, 4, and 6 to Components 1 and 4. This potential for poorly connected topologies resulting from mutation is addressed in component-swap mutation.

Component-swap mutation swaps the position of a flow-mixing or flow-diverting component with another component in the system in a manner that preserves the connectivity of the system. Figure 8 illustrate a system in which the position of a flow-diverting Component 1 is swapped with Component 3 to produce a new system topology. Component-swap mutation is more likely to result in a connected system but has less exploratory power than link-swap mutation. All three topology mutation operators are included in a hyper-mutation operator, with the probabilities of a particular mutation occurring being set to 0.06 for the random-initialization mutation, 0.31 for the link-swap mutation, and 0.63 for the component-swap mutation.

Airflow Rate and Component Capacity Chromosome Mutation. Three mutation operators have been implemented for use with the flow-capacity chromosomes, two of which are established operators with the third being a new problem specific operator. The two established operators are a random mutation and a Gaussian mutation; the problem specific operator is known as
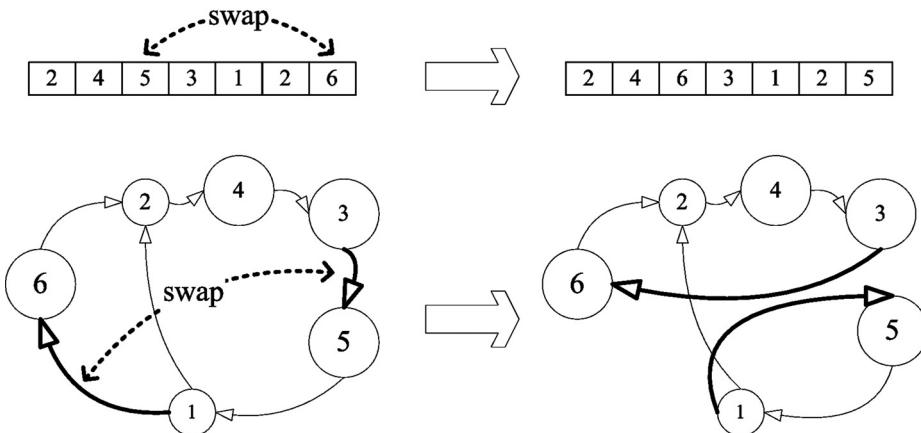

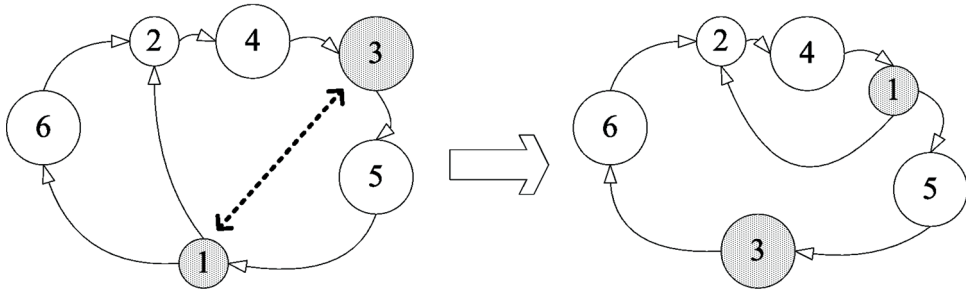**Figure 7. Example link-swap mutation.**

Figure 8. Example component-swap mutation.

the flow-capacity reduction mutation. Random mutation probabilistically changes the problem variable values by a random amount. Gaussian mutation gives more control over this operation by changing the problem variable value according to a Gaussian distribution (the shape of the distribution being defined by a standard deviation).

The flow-capacity reduction mutation is a problem-specific operator that addresses the goal of minimizing the system energy use by reducing the airflow rates by a small fraction and setting randomly selected component capacities to zero (the airflow rates are not set to zero as there will almost certainly be some airflow through most branches of the system).

In a manner similar to the other recombination and mutation operators, the three flow-capacity operators are combined in a hypermutation with the probability of a particular method being used of 0.25 for the random mutation, 0.25 for the Gaussian mutation, and 0.5 for the flow-capacity reduction mutation.

Replacement

A new population of solutions is created by successively replacing the existing solutions with newly recombined and mutated solutions. However, in order to ensure that the search does not converge towards a solution that is inferior to the best solution found to date, the top few percent of the best ranked solutions are carried forward to the next population without any modification (the population overlap being a form of so called "elitism").

ALGORITHM PERFORMANCE

The performance of the EA is examined here in terms of the effect of the different algorithm operators on both the optimality of the solutions and the exploratory power of the search. The performance of the algorithm in terms of the characteristics and optimality of the solutions is examined in an associated publication (Wright and Zhang 2008). Since the EA is a probabilistic search method, its performance has been examined using 16 separate trial optimizations. The results of the trial optimizations are reported as the average of the best solutions found at each iteration (epoch) of the algorithm together with the solutions found on completion of the optimization. The algorithm control parameter values used in this study are given in Table 1. The parameter values have either been taken as the generally accepted values or derived from small-scale trials of the behavior of the algorithm as follows:

• Large population sizes, often in excess of 1000 individuals, are required for an effective search of large topological search spaces (Koza et al. 2003).
• Since the solutions are rank ordered in terms of their fitness, a population of 1000 and a 2% overlap of solutions will result in the top 20 ranked individuals being copied directly to the next generation.

Table 1. EA Control Parameters

Operation	Control Parameter	Value
	Population size	1000
Population	Maximum generations	10,000
	Population overlap	2%
Selection (Tournament Selection)	Tournament type	1-from-2
	Topology probability	50%
Hyper-crossover	Airflow rate and component capacity probability	100%
	Topology probability	2%
Hypermutation	Airflow rate and component capacity probability	10%
Fitness (Stochastic Ranking)	Probability of infeasible solution ranking	45%
Ageing	Maximum evaluations per generation	20

- A 1-from-2 tournament selection provides sufficient selection pressure to move the search towards the fittest solutions without resulting in premature convergence.
- A high probability of crossover is commonly used in topological optimization problems (typically 100%) (Koza et al. 2003). However, a 100% topology crossover was found to be too disruptive, often resulting in infeasible solutions. A lower rate of 50% probability was found to give good algorithm performance.
- Similarly, a high mutation rate on the topology chromosomes was found to result in poor convergence on a feasible solution and, therefore, was assigned a lower value than that for the airflow rate and component-capacity chromosomes.
- The control parameter for the stochastic ranking fitness was taken as that reported by the algorithm's authors (Runarsson and Yao 2000).
- Since the ageing operator (Wright and Zhang 2005) was developed as part of this research, no previous guidelines exist for the choice of control parameter. A comprehensive study of the operator is beyond the scope of this research; however, a value of 20 appears to give good algorithm performance. A value of 20 indicates that any topology having a cumulative total number of solutions greater than 20 per generation will have its fitness penalized. The experimental results indicate that this rate (equivalent to 2% of the population) is sufficient for the topology to evolve.

Example Optimization Problem

The example system used to evaluate the algorithm serves two zones of a building located in Oklahoma City, USA (a location that experiences a largely continental climate). The building and associated load conditions are described in two related publications, Wright et al. (2004) and Wright and Zhang (2008). The annual system operation is represented by nine load conditions within which five different operating regimes occur (Table 2).

Two groups of experiments are performed here, one that includes the optimization of the component set and a second group in which the component set is fixed. For the case in which the component set is optimized, it is difficult to describe the complexity of the problem space since it is a direct function of the component set, this being a variable of the optimization.

Table 2. System Operating Regimes

Zone	
East	**West**
Heating	Heating
Heating	Cooling
Heating	Free cooling
Free cooling	Free cooling
Cooling	Cooling

However, the complexity of the search space can be described for the case when the component set is fixed. The fixed component set used in this study consists of 17 components: two zones, two heating coils, two cooling coils, two humidifiers, four air-mixing sections, four air-diverting sections, and the ambient environment. This results in a topology search space that is in the order of 3.2×10^{19} alternative topologies. The component set also results in 21 discrete variables encoded in a single topology chromosome, six component capacity variables, and five airflow-rate variables for each of the nine boundary conditions (giving 99 continuous variables encoded in nine separate flow-capacity chromosomes).

The number of constraints is also a function of the component set. For the fixed component set experiments, there are ten equality constraints on the feasibility of the topology, one equality and nine inequality constraints on the evaluation of system performance, and 108 inequality constraints on the operation of the system components (for each boundary condition, these constraints relate to the lowest feasible temperature leaving the cooling coils and humidifiers and the supply air temperature and airflow rates entering the zones). Further, for each boundary condition, the quantity of outside air entering each zone is constrained by an inequality constraint, and the supply-air temperature and humidity are constrained by equality constraints (the equality constraints are subject to a small tolerance). Given nine boundary conditions, this results in a total of 54 zone constraints (18 inequality and 36 equality constraints).

In summary, for the fixed component set, there are a total of 120 problem variables (21 discrete and 99 continuous), and the problem is constrained by 135 inequality constraints and 55 equality constraints.

Results

The experimental results are given in Figures 9–11 and Table 3. First consider the results labeled *component set optimized* and *hyperoperators and ageing*. Both of these results are for the full implementation of the algorithm, the difference in the two results being that the component set was optimized for the results labeled *component set optimized* but was fixed for the results labeled *hyperoperators and ageing*. Optimizing the component set results in average infeasibility and objective function values that are of the same order of magnitude as when the component set is fixed. However, after 10,000 iterations (epochs) of the algorithm, only 19% of the solutions were feasible when the component set was optimized, as compared to 81% when the component set was fixed (Table 3). This characteristic is mirrored in Figure 9, which shows the mean infeasibility of the best solutions in all 16 trial optimizations at each iteration (epoch). There is little difference between the results for the early generations, but the mean infeasibilities begin to diverge at approximately 100 generations, with the optimization for the fixed component set (the *hyper and ageing* curve) ultimately having the lowest infeasibility. Note that, for all infeasible solutions found here, the infeasibility is largely due to the system being unable to meet the required supply-air conditions at one or more load conditions. This characteristic
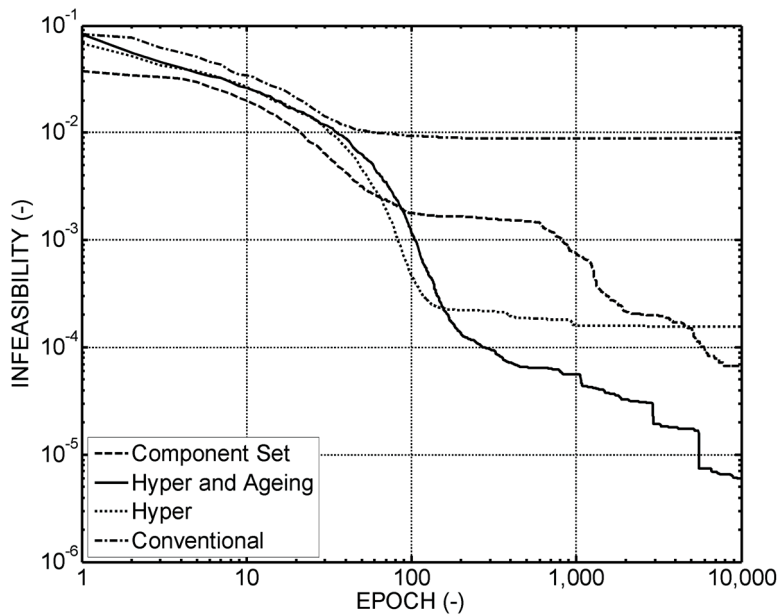
Figure 9. Solution infeasibility.

Table 3. Value of Best Solutions at the Final Epoch

Performance Metric	Component Set Fixed			Component Set Optimized
	Conventional Operators	Hyperoperators	Hyperoperators and Ageing	
Probability of Feasibility, %	6	44	81	19
Mean Infeasibility of Infeasible Solutions, –	9.4×10^{-3}	2.8×10^{-4}	3.2×10^{-5}	8.3×10^{-5}
Mean Objective, kW	8.6	12.6	8.6	11.5

maybe a function of the equality constraints on the supply-air conditions. In this respect, an alternative approach that does not require the use of equality constraints has been suggested (Sowell 2002) but not implemented.

Optimization of the component set results in a significantly lower objective function value during early generations (compare the *component set* and *hyper and ageing* curves in Figure 10). This characteristic suggests that the possible cause of the low probability of finding a feasible solution during the component-set optimization may be due to the low objective function values in the early part of the search overdominating the search direction and preventing the search from finding feasible solutions (which probably requires higher energy use). Given the low probability of finding a feasible design solution when the component set is optimized, the decision was made to fix the component set in all subsequent optimization studies performed during this research. The principle developed in fixing the component set was that there should be a sufficient number of components of each type to allow the synthesis of dedicated subsystems for
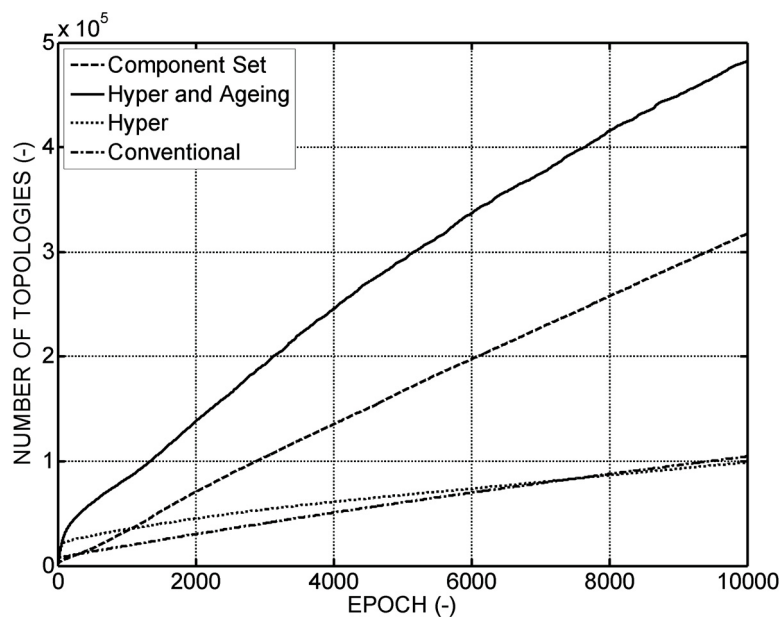
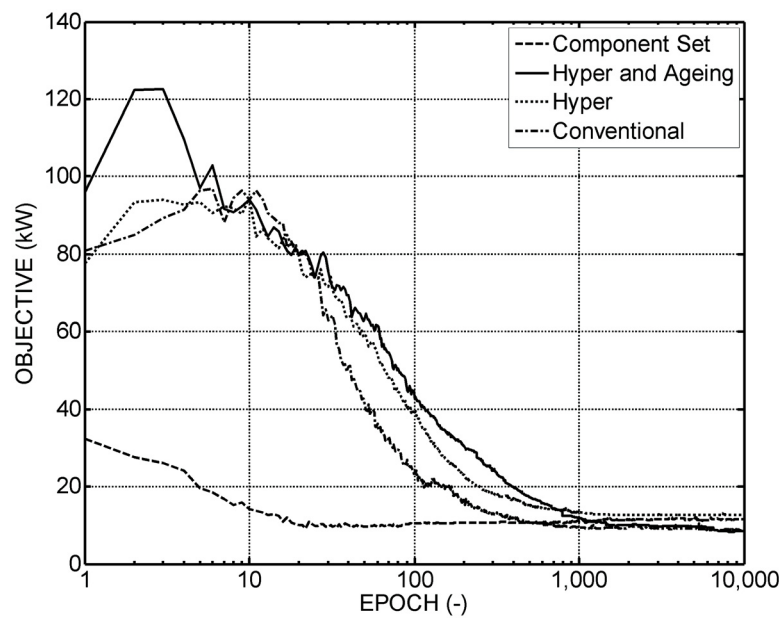**Figure 10. Number of topologies explored.**

Figure 11. Solution objective function value.

the conditioning of each zone; that is, the number of heating coils, cooling coils, and humidifiers should be no less than the number of zones. Several air-mixing and air-diverting sections should also be included to allow the synthesis of suitable air-mixing paths.

Having fixed the component set, further experiments were performed to examine the effect of the problem-specific and ageing operators on the performance of the search. Three sets of experiments were performed, one for the algorithm configured to use conventional search operators; a second set with the algorithm configured to use the hyper-recombination and mutation operators; and finally, a third set with the algorithm using both the hyperoperators and the ageing operator. The conventional recombination and mutation operators used here are the N-point crossover for the topology chromosomes, blend crossover for the flow-capacity chromosomes, and a random mutation for both the topology and flow-capacity chromosomes.

Table 3 indicates that the most significant effect of the hyperoperators and ageing is on the probability of finding a feasible solutions, which increases from just 6% for the conventional search operators to 81% for the hyperoperators with ageing. Further, Figure 9 illustrates that very little improvement in the mean infeasibility of the best solutions is made after the first 100 generations when conventional search operators are used. Conversely, the hyperoperators and ageing allow the search to continue towards more feasible solutions, the highest rate of reduction being for the hyperoperators with ageing.

Figure 10 illustrates the increase in the number of topologies searched with each epoch. The most significant result is the effect of the ageing operator in increasing the number of topologies searched, which is responsible for the high probability of finding a feasible solution (that is, a topology that can provide the required supply-air conditions). Surprisingly, optimizing the component set reduces the number of alternative topologies searched, due perhaps to changes in the component sets restricting the development of a search through a particular subset of topologies.

Finally, for the fixed component set optimization, the behavior of the search in the objective function domain was generally unaffected by the choice of different search operators (Figure 11).

It could be concluded that the development of problem-specific search operators implemented as part of a probabilistic hyperoperation together with the new ageing operator significantly improves the performance of the optimization, particularly in terms of the probability of finding a feasible solution. However, the algorithm was unable to simultaneously optimize the component set, topology, and system operation, which resulted in the use of a fixed component set in subsequent optimization studies (Wright and Zhang 2008).

CONCLUSIONS

The synthesis of novel HVAC system configurations can be formulated as a model-based numerical optimization problem. There are three suboptimization problems associated with design of HVAC system configurations: the selection of the component set, the design of the topological connections between the components, and design of a system control strategy. The approach adopted in this research is to simultaneously solve all three suboptimization problems.

In an attempt to reduce the computation time associated with simulating the system performance, the component performance models have been simplified and the solution of the system performance equations integrated into the optimization problem (as a set of equality constraints on the supply-air conditions). Further, rather than attempt to automatically design a novel control system for each new synthesized system configuration, the control variables have been incorporated directly into the optimization, such that the optimization algorithm acts as an optimum controller.

The optimization problem has been solved using an EA—in particular, the GA subclass. GAs lend themselves to the solution of system synthesis optimization problems in that they can be configured to solve multimodal, nonlinear, highly-constrained, topological optimization prob-

lems of high dimensionality. In particular, GA can be designed to have separate data structures and search operators for each of the optimization subproblems (component set, system topology, and system operation). Designing and implementing separate problem-specific search operators for each subproblem increases the effectiveness and robustness of the algorithm. In this research, several such problem-specific search operators have been implemented. A new search operator was also developed to increase the effectiveness of the algorithm in exploring the topological search space.

The performance of the algorithm has been examined here in terms of the effect of the various search operators on the performance of the algorithm, particularly with respect to the probability of finding a feasible design solution. The performance of the algorithm and optimality of the synthesized HVAC systems is further examined in an associated publication (Wright and Zhang 2008). It can be concluded that the algorithm is unreliable in finding feasible design solutions when the component set, system topology, and system operation are simultaneously optimized. However, when the component set is fixed (as a boundary condition on the optimization) then the algorithm is able to find feasible system configurations with an 81% probability.[3] The probability of finding feasible solution was significantly improved by the design of the problem-specific and novel search operators. It can be concluded that, given a candidate set of system components, the algorithm provides a suitable tool for exploring the novel design of HVAC systems.

ACKNOWLEDGMENT

REFERENCES

Angelov, P.P. 2001. Supplementary crossover operator for genetic algorithms based on the centre of gravity paradigm. *Control and Cybernetics* 30(2):159–76.

Bäck, T. 1996. *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. Oxford: Oxford University Press.

Caldas, L.G., and L.K. Norford. 2002. A design tool based on a genetic algorithm. *Automation in Construction*. 11:173–84.

Caldas, L.G., and L.K. Norford. 2003. Genetic algorithms for optimization of building envelopes and the design and control of HVAC systems. *Journal of Solar Energy Engineering.* 125(3):343–51.

De Jong, K.A. 2006. *Evolutionary Computation: A Unified Approach.* Cambridge, MA: The MIT Press.

Deb, K.D. 2001. *Multi-Objective Optimization using Evolutionary Algorithms.* John Wiley and Sons Ltd.

Diwekar, U.M., I.E. Grossman, and E.S. Rubin. 1992. An MINLP process synthesizer for sequential modular simulator. *Industrial & Engineering Chemical Research* 31:313.

Gero, J.S., and V.A. Kazakov. 1998. Evolving design genes in space layout planning problems. *Artificial Intelligence in Engineering* 12:163–76.

Goldberg, D.E. 1989. Genetic Algorithms in Search Optimization and Machine Learning. Boston: Addison-Wesley.

Jo, J.H., and J.S. Gero. 1998. Space layout planning using an evolutionary approach. *Artificial Intelligence in Engineering* 12:149–62.

Koza, J.R. 1992. Genetic Programming: On the Programming of Computers by the Means of Natural Selection. Cambridge: The MIT Press.

Koza, J.R., M.A. Keane, M.J. Streeter, J. Mydlowec, and G. Lanza. 2003. *Genetic Programming IV: Routine Human-Competitive Machine Intelligence*. Norwell, MA: Kluwer Academic Publishers.

Michalewicz, Z., and D.B. Fogel. 2000. *How to Solve It: Modern Heuristics*. New York: Springer-Verlag.

---

3. In a related study (Wright and Zhang 2008), 66% of the solutions where found to be feasible. This value is based on 41 trial optimizations, whereas only 16 trail optimizations were used here.

Nassif, N., S. Kajl, and R. Sabourin. 2005. Optimization of HVAC control system strategy using two-objective genetic algorithm. *HVAC&R Research* 11(3):459–86.

Runarsson, T.P., and X. Yao. 2000. Stochastic ranking for constrained evolutionary optimisation. *IEEE Transactions on Evolutionary Computation* 4(3):284–94.

Sowell, E. 2002. Oral Communication. See Appendix F of Wright et al. (2004).

Wang, W., R. Zmeireanu, H. Rivard. 2005. Applying multi-objective genetic algorithms in green building design optimization. *Building and Environment* 40(11):1512–25.

Wetter, M., and J.A. Wright. 2004. A comparison of deterministic and probabilistic optimization algorithms for non smooth simulation-based optimization. *Building and Environment* 39:989–99.

Wright, J.A., and V.I. Hanby. 1987. The formulation, characteristics, and solution of HVAC system optimized design problems. *ASHRAE Transactions* 93(2):2133–145.

Wright, J.A. 1996. HVAC optimisation studies: sizing by genetic algorithm. *Building Services Engineering Research and Technology* 17(1):7–14.

Wright, J.A., and R. Farmani. 2001. The simultaneous optimization of building fabric construction, HVAC system size, and the plant control strategy. *Proceedings of the 7th IBPSA Conference: Building Simulation, Rio de Janeiro, Brazil*, 2:865–72.

Wright, J.A. and Y. Zhang. 2005. An ageing operator and its use in the highly constrained topological optimization of HVAC system design. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO), Washington, DC,* pp. 2075–85.

Wright, J.A., and Y. Zhang. 2008. Evolutionary synthesis of HVAC system configurations: Experimental results. *HVAC&R Research* 14(1).

Wright, J.A., H.A. Loosemore, and R. Farmani. 2002. Optimization of building thermal design and control by multi-criterion genetic algorithm. *Energy and Buildings* 34(9):959–72.

Wright, J.A., Y. Zhang, P.P. Angelov, R. Buswell, and V.I. Hanby. 2004. Building *System Design Synthesis and Optimization*. RP-1049 Final Report, American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc., Atlanta.

Zhang, Y. 2005. Synthesis of optimum HVAC system configurations by evolutionary algorithm. PhD thesis, Loughborough University, Loughborough, Leicestershire, UK.