Multiple Evolutionary Agents for Decision Support

K-M Chao[+], C.Laing[#], R. Anane[+], M. Younas[+], P. Norman[*]
[+] *Distributed Systems and Modelling Research Group,*
School of Mathematical and Information Sciences, Coventry University, UK
E-mail: {k.chao; r.anane;m.younas}@coventry.ac.uk
[#]*Technology Faculty, Southampton Institute, Southampton, UK*
Email: christopher.laing@solent.ac.uk
[*]*Engineering Design Centre, University of Newcastle-upon-Tyne, UK*
E-mail: peter.norman@newcastle.ac.uk

In a complex engineering design, multiple engineering design teams are required to work cooperatively in order to design a single product. However multiple disciplinary teams in different departments and companies often use different systems with different modelling methods and techniques. While the parallelism of this approach can reduce the design life cycle, the complexity of the optimisation problem can lead to difficulties. Agent technology can alleviate some of the difficulties of concurrent design. In this paper, we present an architecture in which agent technology is used to model interactions between design systems, and thereby enable design agents from various design disciplines to explore the design problem and solution space. We also introduce a multiple-evolutionary approach with an automated negotiation mechanism to enable agents to exchange design solutions and to reach an agreed optimised solution for a global design problem.

1. Introduction

The design process is often seen as a search for an optimised state in a large space (Simon, 1981). As the design becomes more complex, then the search space becomes larger. A complex engineering design problem often involves multiple disciplinary teams in different departments and in extreme circumstances, multiple disciplinary teams in different departments within multiple companies. However, while the inherent parallelism of the concurrent engineering approach can reduce the design life cycle, the need to work within a set of constraints and the complexity of the optimisation problem can lead to difficulties. The multi-company approach often involves a number of systems with different concerns and perspectives on the problem domain.

Agent technology can alleviate some of the difficulties of concurrent design, albeit within some constraints. Agents tend to specialise in solving one aspect of a problem. Although they only have partial information or knowledge about the overall design problem, they need to meet global requirements in achieving a common goal. In addition, each agent models the problem domain in its own, and different, way. The issues they consider are numerous, and this may lead to mismatches between the issues that other agents are concerned with. However, due in part to their autonomous and proactive nature, agents can overcome these limitations. Although the design activities of different agents are carried out concurrently, the global consistency between their final solutions can be maintained. This implies that local and global constraints and requirements must be satisfied.

2. Engineering Design Requirements

The trend in complex engineering design is towards decentralisation and concurrency. It requires an integrated architecture to co-ordinate concurrent activities. Each participating agent, by playing a certain role and performing a specific function, has its own goal and constraints to meet. To build a common product model requires the agents to co-operate. Multi-disciplinary design may require a conflict resolution mechanism in order to reach an optimised solution and optimisation may be a multi-criteria problem. Moreover, the interdependency between participating agents must be explicitly addressed. Agents can simultaneously and independently evaluate their solution from their own perspective. For example, the case study deals with the design of a heat exchanger. The primary function of a heat exchanger is the transference of heat. However the performance of a heat exchanger in transferring heat is affected by a number of design factors. The optimum solution will require the design agents to work co-operatively. That is, while the operational cost (OC) agent can only contribute to the solutions related to cost, it may require some information from other agents, such as the surface area of the heat exchanger from the operational parameter (OP) agent. However, while the OP agent independently generates optimised values for the velocity of the tube fluid and the surface area of the heat exchanger, it will need to consider the constraints imposed by operational requirement (OR) agent. Indeed the optimum water outlet temperature (OWOT) agent may propose a value for the optimum outlet temperature (t_2) that stimulates the OR agent to produce a better pressure drop calculations. A summary of the requirements that large-scale multiple design agents should meet is given below:

- Decentralised decision-making: each agent can autonomously evaluate its solutions and make decisions. For example, the OR agent is able to design the pressure drop process flow and evaluate its efficiency independently.
- Tolerance of mismatch between issues: each agent tends to solve one aspect in the problem domain. Modelling the problem domain and solutions can take different forms. The system could support these differences through collaboration. For example, the objective of the OR agent is to ensure that the required pressure drop is below 70 k Pa. It has no knowledge about the optimum water outlet temperature calculated by the OWOT agent.
- Generation of a consistent global solution: a single product model should be generated. The data model in each agent should be consistent with models in other agents. For example, the task of the OR agent is to produce a minimum total loss of pressure drop for the tube side. Parameters in the OR agent should be consistently represented in other agents such as the OC agent or the OWOT agent.
- Exchange of ideas: agents explore the search space to find a new solution. For example, the solution generated by the OR agent may be able to motivate the OWOT agent to produce an optimised solution.
- Global optimisation: the final solution does not only satisfy the local objectives and constraints, but it also meets the global requirements.
- Co-operation among agents: each agent has bounded knowledge, but it contributes to solutions to solve partial problems. Agents require a common platform to allow them to work together.

3. Genetic Algorithms (GA)

Since engineering design is viewed as a state-space search, it is appropriate that researchers apply genetic algorithms to find solution to this complex problem (Goodberg, 1989). Artificial Genetic Algorithms work by manipulating a number of chromosomes. A chromosome contains information about the solution and the problem domain that it represents. A population is formed by the same set of chromosomes. A number of genes constitute a chromosome. Each gene encodes a particular issue that the agent needs to consider and it has its own position in the chromosome.

The process associated with a GA starts with a random generation of a set of possible solutions and problem domains, as chromosomes. A function is required to evaluate the fitness of each chromosome in the population, as a solution. A selection criterion and method are also used to choose parent chromosomes, necessary for reproducing offspring. The operations used to reproduce offspring are crossover and mutation. A crossover is the exchange of some genes between parents. After a crossover is performed, a mutation, which randomly alters the value of genes in the new offspring, may take place. These processes will continue until terminating conditions are met. The new offspring will replace the worst performing chromosomes in the population in order to keep the best solutions. This artificial evolutionary approach has proved an effective state search technique in a large and complex space.

A multi-evolutionary approach takes this approach one step further by combining two or more GA systems that represent different problem domains and solutions. During multi-evolution, the systems exchange chromosome or genes in order to produce an effective solution. The main benefit of this approach is that each GA does not only retain its own evaluation function and system properties, but it also communicates with its counterpart. A system can, however, take decisions independently. Researchers who have applied this approach in complex search spaces have shown that this approach outperformed those based on a single GA system (Kauffman, Macready and Dickinson, 1994).

4. Structure of the Proposed Architecture

Figure 1 shows the proposed architecture, indicating the relationships among four design systems. Each agent (representing a design system) can communicate with other agents. Each agent contains a genetic system to evolve its solutions and to evaluate them internally. The communication between agents is through mutual exchange of solutions and through negotiation (Chen, et al 2002a; Chen, et al, 2002b). Under this scheme, the population contains a number of chromosomes that represent the possible problem scenario and solutions. In each agent the system generates solutions by taking the parameters encoded in the chromosomes. The evaluation function takes the global and local objectives and constraints into account in the calculation of fitness. Offspring within an agent are generated by means of local crossover and mutation operations. While mutation is an operation confined internally to agents, on a wider scale the global crossover operation takes place between agents. The negotiation process involves offers, counter-offers, agreements and rejections, four options that ensure that agents obtain coherent solutions.

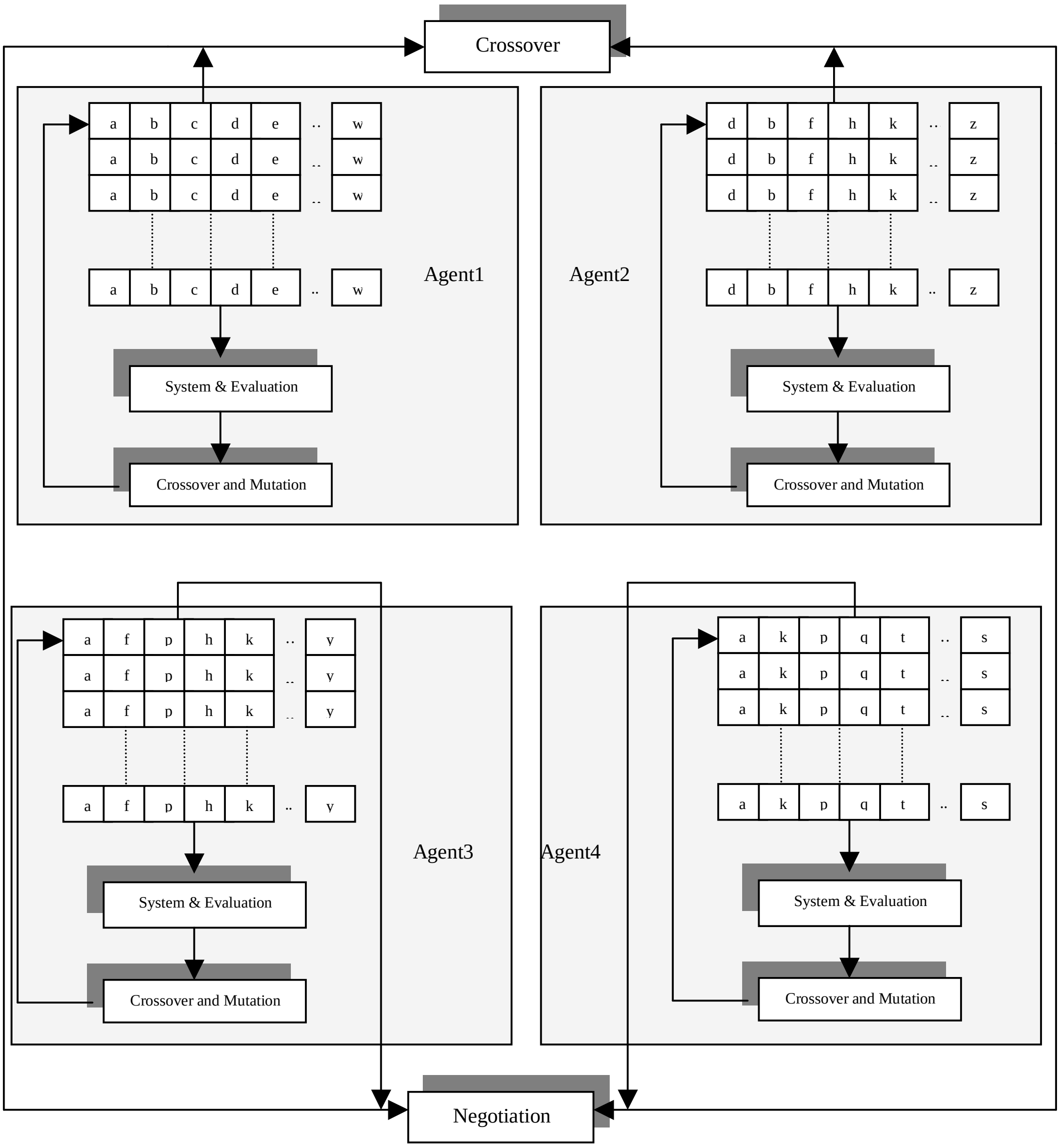


Figure 1 A multiple evolutionary design system

The system operates within a set of constraints and conditions, detailed below:

- An agent has a mechanism that can interpret its chromosomes to produce solutions and is able to evaluate them.
- An agent is aware of those issues shared with other agents. Only agents with shared issues can exchange solutions.
- All agents are working on different areas, so the issues they are concerned with are diverse.
- An agent needs to comply with its hard constraints (i.e., constraints that cannot be violated) in order to achieve its objectives.
- An agent is aware of the global objectives that are related to its local objectives.
- An agent is able to maintain the semantics of chromosomes when a crossover or a mutation operation takes place.

5. Underlying Processes in the Architecture

There are two main processes that are supported by the proposed architecture. The first process is concerned with the exploration of the search space in order to find a possible solution. The second process involves multiple agent negotiation in order to reach an agreement and thereby ensure that an overall coherent solution is produced. A number of issues related to the processes are presented below.

- Each agent initially randomly generates solutions to form a population. The population contains a pool of possible problem scenarios and solutions. The size of the population could vary according to an individual agent's requirements. If the size is large the system cannot work efficiently. If the size is too small, the best solutions could be ignored. Thus, a number of trials need to be carried out by each agent in order to determine an appropriate size. The position of each gene in the chromosome is fixed throughout the process.
- Each solution in the population is interpreted and evaluated by the system in order to determine how well the solution performs. The problem domain and solutions encoded in chromosomes are represented in a bit format. Each gene may contain a number of bits in order to exhibit its properties. The global and local objectives and constraints are represented in the evaluation function. If the objective function is to maximise the utility, then any solution violating the constraints is given a low utility. This will ensure that unsuitable solutions are not selected at a later stage. When a solution is generated, a single fitness value is assigned.
- The tournament approach is used to determine the best two solutions, which are then designated as parents for breeding the next generation. The worst performing chromosomes will be replaced by new offspring.
- Mutation and crossover with different possibilities are introduced in order to explore the search space. An agent explores its local space by only operating on its population.

Agents explore a wider search space by exchanging their solutions. Each agent randomly crosses over its best solution with other agents. Let Agent1, Agent2, Agent3, and Agent4 be four different and interdependent agents that have a number of shared issues. I1, I2, I3, and I4 represent issues that Agent1, Agent2, Agent3, and

Agent4 need to investigate respectively. However, all issues are different, a fact that is denoted by the expression, $I1 \neq I2 \neq I3 \neq I4$. There is no total overlap between issues of any two agents, for example $I1 \not\subset I2$ and $I2 \not\subset I1$or $I2 \cup I1 \neq I1$ and $I2 \cup I1 \neq I2$. If Agent1 and Agent2 have common issues, this can be denoted as $\Phi = I1 \cap I2$ and $\Phi \neq \{\}$; Φ is not an empty set. Any two or more agents can only exchange ideas when they have common issues. In other words, the crossover between two or more agents will be the values of Φ and the issue has to be the same. In the example in Figure 1, Agent1 and Agent2 share two issues, *b* and *d.* Agent2 and Agent3 have common issues *a, f, h,* and *k*. Agent1 and Agent2 can only crossover *b* and *d,* and Agent2 and Agent3 can crossover *a, f, h,* and *k*. Since the crossover between agents is a random process, Agent1 and Agent2 can start, followed by Agent2 and Agent3. Similarly, Agent1 and Agent3 can crossover since they also have shared issues. A randomised crossover approach is introduced because the evolutionary process is non-deterministic. The solution produced at any stage by one agent can lead other agents to explore the space. However, the agents can only crossover the same genes. For example, Agent1 crosses over gene *b* with gene *b* of Agent2, but not with other agents. Once new offspring are generated, the evaluation of the new solutions is performed.

Since each agent has various processing time with their searching, the global crossover operation between agents needs to be synchronised. Each agent has produced at least a satisfactory local solution and met the corresponding global requirement. The negotiation process starts in order to produce a consistent global solution. The negotiation process starts with randomly choosing two agents. For example, Agent1 (beginning with its best-fitted chromosome) makes an offer to Agent2. The agents will evaluate the offered proposal by replacing the value of their shared genes with the proposed ones. If the utility is higher than the threshold, then Agent2 accepts, and an agreement is reached between Agent1 and Agent2. If Agent2 rejects the proposal, Agent 2 must make a counter offer to Agent1. The evaluation process begins again and if the utility is higher than the threshold, then Agent1 accepts the counter offer. If no agreement is possible Agent1 rejects counter offer made by Agent2. Agent1 will then select its second best-fitted chromosome for the next offer. The agents will iterate through this process until either (i) an agreement is reached, or (ii) there is no chromosome that could satisfy their own local objectives and constraints. If this happens the evolutionary process will start again.

When there is an agreement between Agent1 and Agent2, the agreed solution will be presented to Agent3 or Agent4. Agent3 will try to find a solution that matches the proposed solution. If Agent3 runs through all the solutions in its pool and no agreement is reached, then the negotiation process starts again. This process will continue with all agents until the stop criteria are met. If no solution can be agreed upon or the time constraint is broken, then the system stops and concludes there is no solution. If there is an agreement among the agents before the expiry time, the system stops, and a feasible solution is proposed.

The following algorithm represented in pseudo code shows the main procedures of the proposed architecture.

```
Step 1
        For all agents in the set of i
                Agent(i).mutation
                Agent(i).crossover
                Agent(i).evaluation
                Agent(i).selection
                Agent(i).nextgeneration
        End for
    If all agents(i) achieve their local objects and satisfy their local constraints
        then go to negotiation(neg)
        i, k = Randomselection(i) and i!= k
        GlobalCrossover(i, k)
        If timeout()!= ture then Go to step 1
End of Step 1

Step Neg
```

```
             k = RandomSelection(i)
            AgreeParties.add(k)
Lable 1: j = RandomSelection(i)
            If j is in AgreeParties then go to Lable 1
            c=0
            Step Neg1
                        Agent(k).offer = agent(k).select(population(c))
                        If AgreePaties.no > 1 then Agent(j).offer = AgreePaties.deal
                        If agent(k).evaluate(agent(k).offer ) < agent (k).threshold and timeout!=true then go to Step 1
                        c=c+1
                        Agent(k).offerto(j) = Agent(k).offer
    If agent(j).evaluate(agent(k).efferto(j)) > agent(j).threshold
       then agreeparties.deal.=agent(k).offerto()  and AgreeParties.add(j)
            If AgreeParties = all agents
            then  there is an agreed solution and end of program
       else
                                    If noagreement = true and timeout = true then there is no agreed final solution and end of
                                        program
    If AgeeParties < 2 then
       Swap(j, k) and go to Neg1
    Else
        Go to Neg1
            End of Step Neg1
End of Step Neg
```

In the following section, a simple case study will illustrate the concept of the proposed architecture.

6 Case Study: Multiple Heat Exchanger Design Systems

A heat exchanger allows heat to be transferred between fluids without fluid mixing or fluid contact. The most common type of heat transfer equipment used in industry consists of one or more tubes within a shell, and is known as a shell-and-tube heat exchanger. One fluid flows through the tubes, the other through the outer shell and over the outer surface of the tubes. An exchange (or transference) of heat will occur if a temperature difference exists between the two fluids.

If for the hotter fluid, we let Δh_1 represent a *decrease* in enthalpy per kg, and $\dot{m}_1$ represent the mass flow rate per unit time, then $\dot{Q}_1$ (the rate of heat flow *from* the hotter fluid *into* the colder fluid) is represented as follows (Hannah and Hillier, 1999):

$$\dot{Q}_1 = \dot{m}_1 \Delta h_1 \qquad (1)$$

And if for the colder fluid, we let Δh_2 represent an *increase* in enthalpy per kg, and $\dot{m}_2$ represent the mass flow rate per unit time, then $\dot{Q}_2$ (the rate of heat flow *into* the colder fluid *from* the hotter fluid) is represented as follows (Hannah and Hillier, 1999):

$$\dot{Q}_2 = \dot{m}_2 \Delta h_2 \qquad (2)$$

Note heat exchangers are perfectly lagged; consequently the heat flow rate *from* the hotter fluid must equal the heat flow rate *into* the colder fluid, therefore (Hannah and Hillier, 1999):

$$\dot{Q}_1 = \dot{Q}_2 \tag{3}$$

or

$$\dot{m}_1 \Delta h_1 = \dot{m}_2 \Delta h_2 \tag{4}$$

Since enthalpy is given by (Hannah and Hillier, 1999):

$$h = C_p \Delta t \tag{5}$$

where C_p represents the specific heat capacity and Δt represents the temperature drop, then equation (4) becomes:

$$\dot{m}_1 C_{p1} \Delta t_1 = \dot{m}_2 C_{p2} \Delta t_2 \tag{6}$$

Equation (6) represents the heat balance and is usually the beginning of the design process (Hannah and Hillier, 1999).

This heat exchanger case study represents a simplified design problem, but will demonstrate how an automated negotiation mechanism will allow agents to exchange design information and to reach an agreed optimised solution for a global design problem. In this respect it should be noted a heat exchanger's performance (with respect to heat transfer) is affected by a number design factors that interact with each other. For this case study it is assumed that the most important are the surface area of the exchanger, pressure drop throughout the system, velocity of the tube fluid, rate of flow and differences in temperatures of the working fluids. It is further assumed that any improvements to the performance of a heat exchanger requires a balance between the operational factors outlined above, and other demands such as operational costs.

The Problem: Sizing a Heat Exchanger to perform a given task. For example, given inlet temperatures, flow rates, desired outlet temperatures, safety and environmental considerations, and costs (both construction and maintenance), what size heat exchanger is required to perform the task, subject to the following conditions (Janna, 1998)?

The agents need (i) to minimise size of heat exchanger (this requires large flow rate, consequently less surface area and so a smaller heat exchanger can be used), and (ii) to minimise total cost of heat exchanger per year (this requires a small flow rate and consequently a greater surface area, and so a larger heat exchanger must be used). The problem variables are presented below. Four participating agents (using these variables) will collaborate to determine the specifications of a heat exchanger that will meet the global and local design requirements. Table 1, 2, 3 show the default properties of crude oil, water, and design.

Table 1. Crude oil properties

Property	Value	Units
p_o (Density)	786.4	kg/m^3
u_o (Dynamic Viscosity)	0.00189	N s/m^2
C_{po} (Specific Heat Capacity)	2177	J/kg K

$\dot{m}_o$ (Mass Flow Rate)	63.77	kg/s
T_1 (Inlet temperature)	200	C
T_2 (Outlet temperature)	120	C
Δt_o	80	C

Table 2. Water properties

Property	Value	Units
p_w (Density)	995	kg/m^3
u_w (Dynamic Viscosity)	0.00072	N s/m^2
C_{pw} (Specific Heat Capacity)	4186.8	J/kg K
t_1	80	C

Table 3. Design properties

Property	Value
(U) Surface Area Constant	601
(K_h) Header Loss Constant	1.6

The tube ID is encoded in the chromosome, through the evolutionary process will identify a suitable size, but the generation has a min limit of 12 mm. The chromosome also includes the number of tubes and passes, but again is subject to some constraints. The number tubes must be a whole number (e.g., 1, 2, 3, etc.,), the number of passes must lie between 1 and 5, and the number of tubes must be greater than 90.

Once the exchange parameters are selected, calculations are made to determine outlet temperatures, correction factor, and pressure drops. Table 4 shows a set of random data generated for design parameters. The next section will demonstrate how these parameters and values can be used for the agents to calculate their desired properties.

Table 4. Example of Random Generation

Property	Value	Units
t_2 (Outlet water temperature)	100	C
d_i (Tube ID)	14.83	mm
N_t (Number of tubes per shell)	744	

N_p (Number of passes)	4	
E_l (Exchanger length)	5	m

There are four participating agents in the exercise. Each agent concentrates on its pursuing its own objectives and meeting its own constraints. The objective or evaluation functions are different from each other. The representation of chromosomes is different, due to their interest.

6.1 Agent 1: Operational Requirements (OR Agent)

The objective of this agent is to ensure that the pressure drop of the tube side of the heat exchanger should be at a minimum and not greater than 70 k Pa. The variables [i.e., water inlet temperature (t_2); no of tubes per shell (N_t); no of passes (N_p); exchange length (E_l); tube ID (d_i)] will influence these pressure drop calculations. This agent needs to find the best value combination of the above variables in order to produce an optimised solution. The agent chromosome is represented as follows.

t_2	d_i	N_t	N_p	E_l

The table below shows the relationship between the input variables and the objective function.

Table 5: Agent 1 input variables and objective function

Item	Unit	(Saunders, 1988)	
Heat Balance Q(oil) = Q(water)	kJ/s	$\dot{m}_o C_{po} \Delta t_o = \dot{m}_w C_{pw} \Delta t_w$	Crude oil: $\frac{\{63.77 \times (200 - 120) \times 2177\}}{1000}$ = 5275.4 kJ/s Water: $\frac{\{\dot{m}_w \times (t_2 - 80) \times 4186.8\}}{1000}$ = 5275.4 kJ/s With t_2 = 100C, then $\dot{m}_w$ = 63 kg/s
No of tubes per pass (N_{tp})		N_t/N_p	744/4 = 186
Int flow area (one tube) (F_t)	m^2	Area of circle	1.728E-04
Int flow area (one pass) (F_p)	m^2	($N_{tp} \times F_p$)	186 × (1.728 x E-04) = 0.0321

Mass velocity in tubes (m_t)	kg/s m^2	$\frac{\dot{m}_w}{F_p}$	$\frac{63}{0.0321}$ = 1962.62
Reynolds No. (R_e)		$\frac{(m_t \times d_i)}{u_w}$	$\frac{(1962.62 \times 0.01483)}{0.00072}$ = 40424
Friction Factor (f_i)		$0.0035 + \frac{0.264}{R_e^{0.42}}$	$0.0035 + \frac{0.264}{40424^{0.42}}$ = 0.0065
Total Travel (L)	m	$N_p \times E_l$	$4 \times 5 = 20$
Straight Loss (P_i)	Pa	$P_i = \left\{ \frac{4 f_i L m_t^2}{2 p_w d_i} \right\}$	$\left\{ \frac{4 \times 0.0065 \times 20 \times 1962.62^2}{2 \times 995 \times 0.01483} \right\}$ = 67870
Header Loss (P_h)	Pa	$P_h = 1.6 \times \left(\frac{m_t^2}{2 p_w} \right) N_p$	$1.6 \times \left(\frac{1962.62^2}{2 \times 995} \right) \times 4$ = 12387
Total Loss (P_t)	Pa	$P_i + P_h$	80257 = 80kPa Note this breaks condition. Start iteration process again.
The threshold for the total loss is denoted as Pa_t and the utility of Pa is normalised through following: $Pa_r = (Pa - Pa_t)/ Pa_t$. The threshold for the surface area (from Agent 2) is denoted as A_{ot} and the utility of A_{or} is normalised through following: $A_{or} = (A_o - A_{ot})/ A_{ot}$, and the objective function of the OR agent is $U = w_1 \times Pa_r + w_2 \times A_{or}$.			

6.2 Agent 2: Operational Parameter Agent (OP Agent)

The objective of this agent is to ensure that the (i) the correction factor F should be equal to or greater than 0.75, (ii) that the surface area of exchanger should be at a minimum, and (iii) that the velocity of the tube fluid should be within the range 1.5 –3.0 m/s. The variables [i.e., water inlet temperature (t_2); water flow rate ($\dot{m}_w$); number of tubes (N_t); number of passes (N_p)] will influence these parameter calculations. This agent needs to find the best value combination of the above variables in order to produce an optimised solution. The agent chromosome is represented as follows.

t_2	$\dot{m}_w$	N_t	N_p

The table below shows the relationship between the input variables and the objective function.

Table 6: Agent 2 input variables and objective function

Item	Unit	(Saunders, 1988)	
Temperature Factor (*R*)		$R = \frac{T_1 - T_2}{t_2 - t_1}$	$= \frac{200 - 120}{t_2 - 80} = 4$
Temperature Factor (*S*)		$S = \frac{t_2 - t_1}{T_1 - t_1}$	$= \frac{t_2 - 80}{200 - 80}$ =0.167
Temperature Factor (*F*)		$F = \frac{\sqrt{R^2+1}\ln[(1-S)/(1-RS)]}{(R-1)\ln[\frac{2-S(R+1-\sqrt{R^2+1})}{2-S(R+1+\sqrt{R^2+1})}]}$	= 0.95
log MTD	C	$\Delta T_{lm} = \frac{(T_1 - t_2) - (T_2 - t_1)}{\ln\left\{\frac{(T_1 - t_2)}{(T_2 - t_1)}\right\}}$	$\frac{(100-40)}{\ln\left\{\frac{100}{40}\right\}} = 65.48\,\mathrm{C}$
MTD correction factor	C	$\Delta T_m = F \Delta T_{lm}$	0.95 x 65.48 = 62.206
Surface Area (A_o)	m^2	$A_o = \frac{Q}{U_o \Delta T_m}$	$\frac{5275.4 \times 1000}{601 \times 62.206} = 141.11$
Heat Balance Q(oil) = Q(water)	kJ/s	$\dot{m}_o C_{po} \Delta t_o = \dot{m}_w C_{pw} \Delta t_w$	Crude oil: $\frac{\{63.77 \times (200 - 120) \times 2177\}}{1000}$ = 5275.4 kJ/s Water: $\frac{\{\dot{m}_w \times (t_2 - 80) \times 4186.8\}}{1000}$ = 5275.4 kJ/s With t_2 = 100C, then $\dot{m}_w$ = 63 kg/s
No of tubes		N_t/N_p	744/4 = 186

per pass (N_{tp})			
Int flow area (one tube) (F_t)	m^2	Area of circle	1.728E-04
Int flow area (one pass) (F_p)	m^2	($N_{tp} \times F_p$)	186 × (1.728E-04) = 0.0321
Velocity in tubes (V_t)	m/s	$\frac{\dot{m}_w}{p_w \times F_p}$	$\frac{63}{(995 \times 0.0321)}$ = 1.97
The objective function of the OP agent is $U = Min\{ \frac{\dot{m}_w}{p_w \times F_p} \}$			

6.3 Agent 3: Operational Condition Agent (OWOT Agent)

The objective of this agent is to ensure that the water outlet temperature is at the optimum. The variables [i.e., water inlet temperature (t_1); water flow rate ($\dot{m}_w$); water cost (W_c)] will have an influence on the optimum outlet water temperature (OWOT). This agent needs to find the best value combination of the above variables in order to produce an optimised solution. The agent chromosome is represented as follows.

t_2	$\dot{m}_w$	W_c

The following assumptions are required to ensure the agent produces an appropriate optimum solution.
Heat exchanger is in operation 7800 hr/yr (t).
Water costs are between £0.3-£0.6/1000ltr (W_c).
Annual cost of operating exchanger amounts to £20/m^2 of surface area (C_f).
Heat transfer coefficient 142 W/m^2 K (U).
The table below shows the relationship between the input variables and the objective function.

Table 7: Agent 3 input variables and objective function

Item	Unit	(Saunders, 1988; Janna, 1998)	
Heat Balance Q(oil) = Q(water)	kJ/s	$\dot{m}_o C_{po} \Delta t_o = \dot{m}_w C_{pw} \Delta t_w$	Crude oil: $\frac{\{63.77 \times (200 - 120) \times 2177\}}{1000}$ = 5275.4 kJ/s Water: $\frac{\{\dot{m}_w \times (t_2 - 80) \times 4186.8\}}{1000}$ = 5275.4 kJ/s With t_2 = 100C, then $\dot{m}_w$ = 63 kg/s

Temperature Factor (R)		$R=\dfrac{T_1-T_2}{t_2-t_1}$	$=\dfrac{200-120}{t_2-80}$ = 4
Temperature Factor (S)		$S=\dfrac{t_2-t_1}{T_1-t_1}$	$=\dfrac{t_2-80}{200-80}$ =0.167
Temperature Factor (F)		$F=\dfrac{\sqrt{R^2+1}\ln[(1-S)/(1-RS)]}{(R-1)\ln[\dfrac{2-S(R+1-\sqrt{R^2+1})}{2-S(R+1+\sqrt{R^2+1})}]}$	= 0.95
Monetary Units per unit mass	£/kg	$C_w=\dfrac{£W_c(5.19\times10^{-03})}{1000(2.79\times10^{-04})}\dfrac{1}{p_w}$	$C_w=\dfrac{£0.5(5.19\times10^{-03})}{1000(2.79\times10^{-04})}\dfrac{1}{995}$ $C_w=£9.348\times10^{-06}$
Use ‘trial and error’ to find t_2	C	$\dfrac{UFtC_w}{C_fC_{pw}}\left(\dfrac{(T_1-t_2)-(T_2-t_1)}{t_2-t_1}\right)^2=\ln\dfrac{T_1-t_2}{T_2-t_1}-\left(1-\dfrac{T_2-t_1}{T_1-t_2}\right)$	
The objective function of the OWOT agent is U= $Min(\mid(\dfrac{UFtC_w}{C_fC_{pw}}\left(\dfrac{(T_1-t_2)-(T_2-t_1)}{t_2-t_1}\right)^2-\ln\dfrac{T_1-t_2}{T_2-t_1}-\left(1-\dfrac{T_2-t_1}{T_1-t_2}\right))\mid)$			

6.4 Agent 4: Operational Cost Agent (OC Agent)

The objective of this agent is to ensure that the total cost of the heat exchanger per year should be at a minimum, and not greater than £20000 per year. The variables [i.e. water inlet temperature (t_1); Water flow rate ($\dot{m}_w$); Water cost (W_c)] could affect the total operational cost of a heat exchanger per year. The agent needs to find best combination of the values of these three variables in order to produce an optimised solution. So, The chromosome is represented as follows:

t_2	$\dot{m}_w$	W_c

The following assumptions are required to ensure the agent produces an appropriate optimum solution.
Heat exchanger is in operation 7800 hr/yr (t).
Water costs are between £0.5-£0.6/1000ltr (W_c).
Annual cost of operating exchanger amounts to £20/m^2 of surface area (C_f).
Heat transfer coefficient 142 W/m^2 K (U).
The table shows how the relations between input variables and the objective function.

Table 8: Agent 4 input variables and objective function

Item	Unit	(Saunders, 1988; Janna, 1998)	
Heat Balance Q(oil) = Q(water)	kJ/s	$\dot{m}_o C_{po} \Delta t_o = \dot{m}_w C_{pw} \Delta t_w$	Crude oil: $\frac{\{63.77 \times (200-120) \times 2177\}}{1000}$ = 5275.4 kJ/s Water: $\frac{\{\dot{m}_w \times (t_2 - 80) \times 4186.8\ \}}{1000}$ = 5275.4 kJ/s With t_2 = 100C, then $\dot{m}_w$ = 63 kg/s
Monetary Units per unit mass (C_w)	£/kg	$C_w = \frac{£W_c(5.19 \times 10^{-03})}{1000(2.79 \times 10^{-04})} \frac{1}{p}$	$C_w = \frac{£0.5(5.19 \times 10^{-03})}{1000(2.79 \times 10^{-04})} \frac{1}{995}$ $C_w = £9.348 \times 10^{-06}$
Total Cost (C_t)	£	$C_w \dot{m}_w t + C_f A_o$	$C_t = 9.348 \times 10^{-6}(63)(7800 \times 60 \times 60) + 20(141.11)$ = £19358.86
The objective function of the OC agent is $Uoc = Min(C_w \dot{m}_w t + C_f A_o)$			

7. Experimental Result and Analysis

The experiment was set up in a lab with four Pentium III 650MHz PCs in a LAN to accommodate four agents. Each agent has 100 chromosomes in the pool and evolves over 500 generations. The possibility for local crossover is 40% and the possibility for mutation is 50%. The global crossover rate is 100%. The figures 2, 3, 4, and 5 show the individual evolving progress. In order to gain the system efficiency, the tournament approach is used for the selection of the parents, so the solutions were not miscellaneous. However, a possible side effect of this approach could be a solitary solution or if the number of generations is too large a small number of possible solutions. As a result, agents could experience some difficulties in converging to an agreement at the negotiation stage. In order to prevent this from occurring, a number of experiments were carried out to find the appropriate generation numbers and population sizes for the agents. Figure 4 shows the OWOT agent could find a possible optimised solution when the water inlet temperature (t_2) is 125°C, the water flow rate ($\dot{m}_w$) is 28 kg/s and the water cost (W_c) is £0.5 per 1000 litres. The diagram also shows a value of 410, this is the balance of outlet water temperature efficiency; augmented 10000 times. There are other feasible solutions in the population with different combination of values such as 115 °C for the water inlet temperature, 36 kg/s the water flow rate and £0.51 per 1000 litres for the water cost.

A number of iterated processes between evolutions and negotiations were performed among agents. Finally all agents agreed on one solution; 125^{o}C for the water inlet temperature, 28 kg/s for the water flow rate, and £0.5 per 1000 litres for the water cost. As a result, the OWOT agent was able to find an optimised heat balance and the OC agent obtains £11,532 for the minimum operational cost (C_t). When no of tubes per shell (N_t) is 393, no of passes (N_p) is 4, exchange length (E_l) is 5 m, and tube id (D_i) is 0.01483 m, the OR agent realises 58930Pa for its the total loss of pressure drop for the tube side. The OP agent agrees on these input parameters to obtain the velocity of the tube fluid (V_t) with 1.865 m/s, 188.9 m^2 for surface area (A_o), and 0.805 for the correction factor (F).

Due to the random seeding, processing times can vary, however if there is no result after 30 minutes, the system is interrupted. Figure 6 shows 5 different processing times. At each processing time the system successfully generated an agreed solution. Note the number of unsuccessful attempts has not been taken into account. It should be noted that if one agent could not find at least one satisfactory solution, then the remaining agents have to wait. The OR and OWOT had particular difficulties in finding a feasible solution. This has reduced the processing speed. The duplications of the solutions in the population pool also increased agent difficulties in finding an agreement. For example in some situations more than half populations had the same solution. These redundant communications among the agents during the negotiation also added to the processing time.

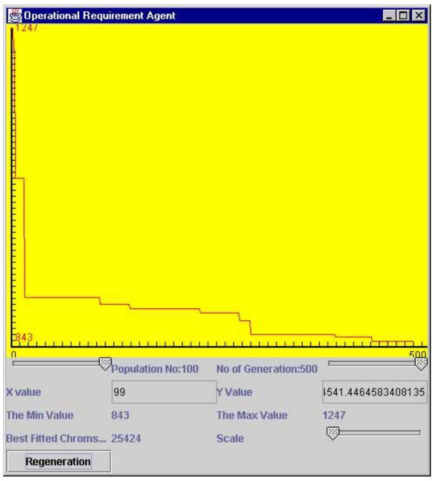


Fig 2. The OR agent evolving progression

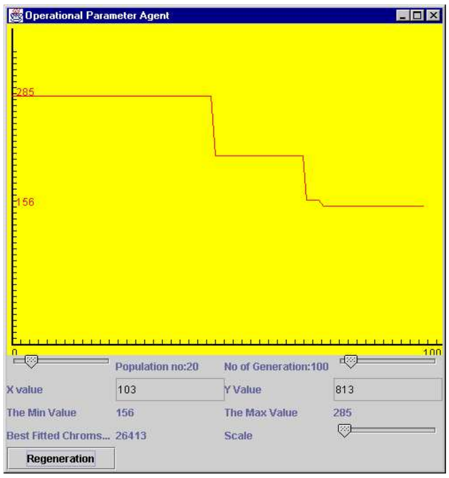
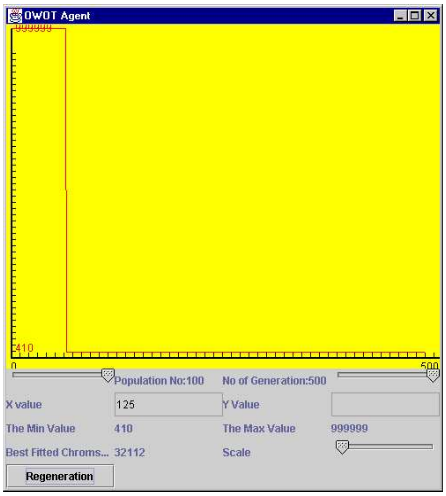


Fig 3 The OP agent evolving progression



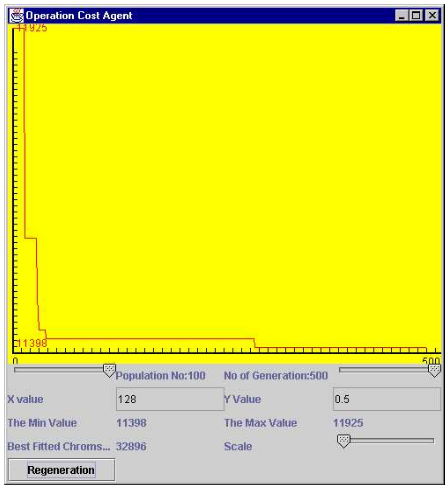


Fig 4 The OWOT agent evolving progression

Fig 5 The OC agent evolving progression

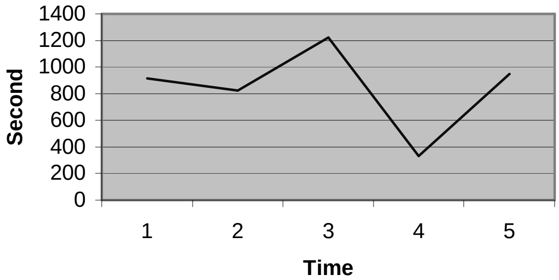


Figure 6 Processing Time in Seconds

8. Discussion and Conclusion

Few researchers have applied co-evolutionary approaches to complex engineering design. A common limitation of the application of co-evolutionary approaches to complex engineering design is that the problem domain does not change throughout the process. This issue was addressed by some authors (Poon and Maher, 1997; Maher and Wu, 1999; Maher and Wu, 1999; Fonseca and Fleming, 1995) who introduced the concept of the evolution to both the problem and the solution in their proposed co-evolutionary approach. One of the main advantages of the co-evolutionary approach is that it may lead to the emergence of new structures and behaviour. However, the problem associated with the experiments conducted in (Poon and Maher, 1997; Maher and Wu, 1999; Maher and Wu, 1999) is that the criteria for fitness change over time and may prevent the system from converging.

In another field of application, one group of researchers introduced a co-evolutionary system to solve some complex aerodynamics problems (Nair and Keane, 2002). Their approach was marked by the preservation of disciplinary autonomy and decentralised decision-making. As a result, the subsystems could operate concurrently and independently. This led to a reduction in software integration and in inter-disciplinary communication overheads, in contrast to a centralised approach. The system is able to accommodate variable-complexity design parameterisation with a mix of discrete and continuous variables and robustness in non-convex design space.

However, the main limitation of the systems mentioned above is that they deal with one issue only. They do not address the problems associated with multiple party negotiations over multiple issues in order to produce a global and coherent product data model.

This paper has presented a new architecture that enables design agents from various design disciplines to explore the design problem and solution space. The design agents in this proposed architecture are able to transfer their design solutions to each other in order to produce a satisfactory product model. There are two components in the architecture, namely multi-evolution and automated negotiation. The multi-evolution mechanism enables agents to exchange design solutions in the design process so as to obtain an optimised product. The automated negotiation mechanism allows agents to generate a coherent solution. The results of the case study demonstrate how effective the proposed architecture can be in resolving multiple objective design problems.

This approach allows each agent to maintain its autonomy in decision-making and to collaborate with other agents in order to generate an optimised design. It is aimed at tackling the complexity of multiple parties negotiating over multiple issues. The shared issues between two parties are likely to be different from issues in other agents. Two agents may have resolved their differences over two or more issues, but this agreement may not be valid when a third party joins the negotiation. One agent may have to change the value of a previously agreed issue in order to reach a new agreement with the third party. The complexity of this process can increase due to the number of issues and agents involved. However, GA is an effective technique for searching a given solution space.

The main drawback of this approach is the requirement to represent the problem domain and the solution in chromosomes. This leads to a design situation that cannot allow for the introduction of new issues or the deletion of existing issues once the chromosomes are formed. The structure of the chromosomes has to be preserved throughout the whole process. This is because the semantics of the chromosomes must be strictly maintained otherwise agents may have difficulties in interpreting them.

Another issue that needs to be addressed is the scalability. We extended the initial case study by developing a more complex scenario. This was achieved by adding more components (such as vessels and pumps etc). To accommodate the new scenario, the high-level problem domain and solutions are encoded in chromosomes. In addition, the representation of the chromosomes needs to change from bits to objects. The object structure is based on the specifications of the STEP AP231 (Owen, 1994). The preliminary result shows that the processing time increases considerably. This leads to a run-time performance that was unacceptable, and which may require high performance or grid computing (Chao, 2002) approach to resolve this issue.

References

Carlos M. Fonseca and Peter J. Fleming, 1995, "An Overview of Evolutionary Algorithms in Multiobjective Optimisation", *Evolutionary Computation*, Vol 3, No1, pp.1-16.

Chao, K-M, Anane, R., Chen, J-H, Gatward, R., 2002, "Negotiating Agent in a Market-Oriented Grid", *Conference Proceedings of the 2nd IEEE/ACM International Symposium on Cluster Computing and the Gird*, IEEE Computer Society Press, pp 436-437.

Chen, J-H, Anane, R, Chao, K-M, Godwin, N., 2002a, "Architecture of an Agent-Based Negotiation Mechanism", *Conference Proceedings of The 22nd IEEE International Conference on Distributed Computing Systems Workshops*, IEEE Computer Society Press, pp379-384.

Chen, J-H, Chao, K-M, Godwin, N., Reeves, C. and Smith, P., 2002b, "An Automated Negotiation Mechanism Based on Co-Evolutionary Approach and Game Theory", *Conference Proceedings of the 2002 ACM Symposium on Applied Computing*, ACM Press, pp63-65.

Goodberg, D.E., 1989, "Genetic Algorithms: In Search of Optimisation and Machine Learning", Addison-Wesley.

Hannah, J. and Hillier, M. J., 1999, *Mechanical Engineering Science*, 3rd edition, Addison Wesley Longman Press.

Janna, W. S., 1998, *Design of Fluid Thermal Systems*, 2nd edition, PWS Publishing Company.

Kauffman, S., Macready, W. G. and Dickinson, E., 1994, "Divide to Coordinate: Co-Evolutionary Problem Solving", *Santa Fe Institute Technical Report*, SFI-94-06-031.

Maher, M. L. and Poon, J., 1996, "Modelling Design Exploration as Co-Evolution", *Microcomputers in Civil Engineering*, Vol. 11, No 3, pp.195-209.

Maher, M.L., and Wu, P. X., 1999, "Reconsidering Fitness and Convergence in Co-Evolutionary Design", *Proceedings 12th Australian Joint Conference on Artificial Intelligence*, Lecture Notes in Artificial Intelligence Vol.1747, Springer-Verlag, Berlin, Germany; pp. 488-489.

Owen, J., 1994, *STEP An Introduction*, Information Geometer Ltd. UK.

P. B. Nair and A. J. Keane, 2002, "Coevolutionary Architecture for Distributed Optimisation of Complex Coupled Systems", A.I.A.A. Journal, Vol. 40, No 7, pp. 1434-1443.
Poon, J. and Maher, M.L., 1997, “Co-evolution and Emergence in Design”, Journal of AI in Engineering, Vol. 11, No 3, pp.319-327.
Saunders, E. A. D., 1988, “Heat Exchangers: Selection, Design & Construction”, Longman Scientific & Technical Press.
Simon, H. A., 1981, “The Science of the Artificial”, 2nd edition, MIT Press.