AGGREGATED BANDWIDTH ALLOCATION: INVESTIGATION OF PERFORMANCE OF CLASSICAL CONSTRAINED AND GENETIC ALGORITHM BASED OPTIMISATION TECHNIQUES

A. Pitsillides¹, G. Stylianou¹, C. S. Pattichis¹, A. Sekercioglu², A. Vasilakos³

Abstract—

We consider a high speed integrated services network, and investigate compare and contrast the performance of two algorithms for solving the Aggregated Bandwidth Allocation (BA) problem. The algorithms we focus our attention are: (1) a classical constrained optimisation (CCO) algorithm and (2) a constrained optimisation Genetic Algorithm (GA). We adopt the Virtual Path (VP) concept for ATM (Asynchronous Transfer Mode) networks as an example, but expect the findings to be equally applicable for other aggregated bandwidth allocation problems, such as in networks using MPLS (Multiprotocol Label Switching). We define a fair multiobjective criterion for maximisation of network throughput, based on a probability (density) function for bandwidth demand. We convert to single optimisation problem and study network topologies involving varying number of nodes. We compare throughput, fairness and time complexity of GA-BAVP and CCO-BAVP. The results on maximising the throughput obtained with GA-BAVP and CCO-BAVP are in close agreement, however when considering fairness GA-BAVP outperforms CCO-BAVP, especially for more complex topologies. Convergence of the two algorithms appears similar, with GA-BAVP outperforming CCO-BAVP in initial stages, and vice-versa for longer time scales. However as problem complexity increases the solution time for the Genetic Algorithm does not increase as fast as the classical constrained optimisation algorithm. A hybrid scheme is also introduced, combining the benefits of both algorithms. It exhibited better overall convergence rate but the same solution as CCO-BAVP. More work is needed to investigate the usefulness of GAs in larger network topologies, as well as to multiobjective bandwidth allocation problems.

Index Terms— Aggregated Bandwidth Allocation, Virtual Path, MPLS, Classical Constrained Optimisation, Constrained Optimisation Genetic Algorithms, Multiobjective Optimisation.

I. INTRODUCTION

High speed Integrated Services Networks (ISNs), including recently proposed Internet architectures and ATM, aim to support various classes of multimedia traffic with different bit rates and quality of service requirements; thus traffic control and resource management are crucial in order to guarantee the desired grade of service. Several mechanisms are proposed in high speed integrated service networks to allocate resources and control traffic, such as bandwidth allocation, buffer management, call admission control, input rate regulation, routing, and queue scheduling.

In order to minimise the complexity of resource allocation, aggregated bandwidth allocation has been suggested. For

example, in order to simplify per-call resource management and routing in Asynchronous Transfer Mode based networks the Virtual Path (VP) concept was introduced [1], [2]. A VP is a pre-established path between an Origin-Destination (OD) pair. A VP aggregates a number of Virtual Circuits (VCs), and can reserve bandwidth for a time span longer than the duration of a specific call. A primary objective of the VP concept was to facilitate fast and simple call setup with minimum signalling requirements, and allocate call bandwidth at the network edge, by considering only the allocated VP bandwidth. For Internet based networks, current trends are emerging to allocate bandwidth (to different classes and users), in order to support Quality of Service provision (see, for example, [3, 4, 5, 6, 7, 8, 9]). This despite the complexity introduced when compared with the current Internet best-effort model. The architecture of various proposed solutions differ in detail, but the underlying model is similar in the sense that network resources need to be allocated and policed. Aggregated bandwidth allocation, without the need for per-session signalling, is expected to be useful for traffic engineering in MPLS based Internet architectures [7, 8, 9], as well as for the classical overlay (IP over ATM) model [7].

In this paper we focus on Asynchronous Transfer Mode based networks, but expect that the findings will also be applicable to the Internet models for aggregated bandwidth allocation to implement traffic engineering [6, 7, 8, 9]. Based on a general bandwidth demand function, derived from an appropriate probability (density) function, we aim to solve the VP bandwidth allocation problem. We firstly present an extension of the objective function used in [28] [29] to provide the bandwidth allocation problem with fairness between the VP's [10], and then investigate a classical constrained optimisation and a Genetic Algorithm method to solve the single objective optimisation problem. We investigate the performance of CCO-BAVP and GA-BAVP, for solving the Bandwidth Allocation for Virtual Paths problem, considering several network topologies (3 to 7 nodes, 2 to 42 ODs, 4 to 84 VPs). We also compare fairness and time complexity of the schemes. Here, we define "fairness" as the deviation of assigned capacity from the capacity required to give 95% satisfaction of utility function for each VP. In earlier works [11], [12], we reported on

¹ Department of Computer Science, University of Cyprus, Nicosia, Cyprus, email: Andreas.Pitsillides@ucy.ac.cy

² Centre for Telecommunications and Information Engineering, Monash University, Melbourne, Australia.

³ Institute of Computer Science, FORTH, Crete, Greece.

Evolutionary Programming (EP) and Genetic Algorithm (GA) based optimisation techniques for the solution of the BAVP (EP-BAVP and GA-BAVP) for the case of a 3-node and a 4-node network topology. In [11] we compared EP-BAVP with a classical constrained optimisation algorithm (CCO-BAVP) for a 3-node problem. We showed that EP-BAVP achieves the same throughput for the case of the sum and of the product of objective functions (worth noting that in [10] we showed that these are particular solutions of the multiobjective formulation). In [12], we extended the problem to 4-node case and investigated further the behaviour of EP-BAVP, GA-BAVP and CCO-BAVP. We showed that GA-BAVP provides better performance than EP-BAVP, and similar performance with CCO-BAVP.

Several studies make use of Genetic Algorithm (GA) based techniques to solve network problems. The motivation behind GA's in nonlinear function optimisation problems is that the problem can be expressed as such that natural evolution, as reported, can provide an attractive paradigm for implementing general nonlinear searches [13], [14]. Due to the broad applicability of GA techniques, a broad application domain exists in solving Telecommunication network problems [15], [16]. Application examples include [17], [18], [19], [20], [21], [22], [23], [24], [25]. Elbaum et al [24] use a GA to design the topology of Local Area Networks (LANs) and Ko et al [25] design Mesh Networks. Shimamoto et al [18] consider call blocking probability as a network constraint and apply GAs for network routing. Not many studies address aggregated bandwidth allocation. Some notable examples follow. Pan and Wang [17] code the traffic distribution to represent each chromosome, and use the average delay, derived from an M/M/1 queueing model, as an optimisation constraint to maximise bandwidth allocation. Taterdtid et al [19], [20] address the network configuration problem (i.e obtain best path for each origin-destination pair), based on the VP concept. They use an M/M/1/K queueing model to derive the blocking probability and the total average packet delay as constraints to maximise total network throughput. Swaminathan et al [21] use GAs to predict the bandwidth-demand patterns to enable better VP management. Hybrid GA's have also been used; e.g. [22], [23] investigate the effective management of isochronous channels in high speed networks.

This paper is organised as follows. In section 2 we outline the problem formulation, in section 3 the GA method is presented. In section 4 several problems are presented, and the solutions offered by the classical and GA methods are discussed. Finally, in section 5 we present our conclusions.

II. PROBLEM FORMULATION AND SOLUTION APPROACHES

A. Background

Bandwidth Allocation for Virtual Paths (BAVP) aims to supply optimal bandwidth assignment to VPs by taking global network considerations into account. It is located at

the higher levels of the control structure: the VP and network levels. It is therefore associated with a "slow" time scale in terms of minutes or tens of minutes. A number of approaches were proposed to solve the BAVP problem. For example, Gerla et al [26] developed an M/M/1 queueing model (assuming independence between the queues) for BAVP aimed at minimising total expected delays. Hui et al [27] formulated BAVP as a non-linear programming model which minimises total usage cost. Herzberg [28], Herzberg and Pitsillides [29] proposed an alternative model for BAVP which uses a network carrier viewpoint and maximises total network throughput. Ohta and Sato [1] propose a state-dependent, dynamic bandwidth allocation scheme which reallocates bandwidth on a call-by-call basis, i.e. upon arrival of a new call request to a BAVP, and Mocchi et al [30] use a periodic approach to reallocate bandwidth to VPs to maximise the network utilisation and achieve a desired call acceptance. The offered traffic in most is characterised by Poisson arrivals, exponential holding time distribution, and constant effective bandwidth. Note that a number of optimisation criteria can be incorporated to formulate a Multiobjective optimisation problem [31] that can also be hierarchically organised [32]. Examples of multiobjective formulations include [33] and [34]. Note that most multiobjective functions are converted to single objective to reduce (considerably) the problem complexity. Also, game theoretic concepts may be used to deal with other issues, such as conflicting objectives, or introducing fairness into the VP allocations [35], [36], [37]. For example, Lazar and co-workers [37] set up a non-cooperative game scenario in which each user attempts to reserve the maximum possible bandwidth for its own VP. The cost function for each user is a tradeoff between the cost to the user for each unit of reserved capacity and the call blocking probability, using the Erlang-B loss function for a Poissonian call arrival process. They solve the single resource (single link) problem and show the existence and uniqueness of the Nash Equilibrium Point, and the convergence rates for the Gauss-Seidel and Jacobi iterative schemes. Also for some results they extend to more general topologies.

B. Multiobjective BAVP model

Consider a (virtual) network consisting of N nodes representing ATM switches, and L transmission links connecting the nodes. We try to find an optimal VP bandwidth assignment, which maximises the total expected network throughput, given the network topology; expected OD traffic loads; and link capacities. We seek to provide for "fair" allocations of bandwidth among all VP's (note that different VP's can have different performance objectives). The measure of fairness employed here is based on the concept of Pareto optimality from games theory [38] (also known as efficient, non-inferior and non-dominated optimality)

which applies to co-operative game situations (rather than Nash optimality which applies to non co-operative ones, as used for example in [37]).

We define [10]:

C_i^{link} - Available bandwidth of link i , $i=1,...,L$ for VP assignment.

N_p - Number of network unidirectional OD pairs, indexed $j=1,..., N_p \leq N(N-1)$.

P_j - Number of predetermined possible paths connecting OD pair j (allows for multiple VP's between an OD pair).

$U_{j,p}$ - Bandwidth assigned to OD pair j through path p , $j=1,...,N_p$, $p=1,...,P_j$.

U_j - Bandwidth assigned to OD pair j . Clearly

$$U_j = \sum_{p=1}^{P_j} U_{j,p}, \quad j=1,...,N_p.$$

U^* - Pareto optimal solution, $U^*=[U_1^*,...,U_{j_p}^*]$.

$D_j(U_j)$ - Expected throughput of OD pair j when it utilises bandwidth assignment of size U_j (typically a concave non-decreasing function).

$T_j^{\min}(T_j^{\max})$ - Minimal (maximal) bandwidth assigned by user to OD pair j (e.g. $T_j^{\min}$ can be set to meet minimum performance objectives and $T_j^{\max}$ for fairness).

$\delta_{j,p}^i$ - a (0,1) indicator variable which takes the value of 1 if path p of OD pair j uses link i .

$F_j(U_j)$ ($f_j(u)$) - Probability (density) function for bandwidth demand.

Observe that the $U_{j,p}$ can be seen as the references to be provided to the lower levels.

The mathematical formulation is:

$$\max_U \{D_1(U_1), ..., D_j(U_j)\} \quad (1)$$

Subject to the constraints

$$\sum_{j=1}^{N_p} \sum_{p=1}^{P_j} \delta_{j,p}^i U_{j,p} \leq C_i^{link}, \quad i=1,...,L$$

$$T_j^{\min} \leq U_j = \sum_{p=1}^{P_j} U_{j,p} \leq T_j^{\max} \quad j=1,...,N_p$$

$$U_{j,p} \geq 0, \quad j=1,...,N_p, \quad p=1,...,P_j.$$

Note that $U^* \in U$ the set of all Pareto optimal solutions if and only if $D_j(U_j) \leq D_j(U_j^*)$, $j=1,...,N_p$, with strict inequality for at least one j .

To solve above model, statistical characteristics of functions $D_j(U_j)$, $j=1,...,N_p$ should be known. We assume that each function $D_j(U_j)$ is derived from an appropriate probability function $F_j(U_j)$ for bandwidth demand and a corresponding

probability density function $f_j(u)$. By considering throughput as "fluid flow", function $D_j(U_j)$ is obtained [29]:

$$D_j(U_j) = \int_0^{U_j} u f_j(u) du + U_j \int_{U_j}^{\infty} f_j(u) du = \int_0^{U_j} u f_j(u) du + U_j [1 - F_j(U_j)] \quad (2)$$

The first term is expected throughput for bandwidth demand up to assigned bandwidth of U_j , and second is for demand above assigned bandwidth of U_j . In [29], we show that expected throughput decreases, as variance of bandwidth demand (derived from normal probability) increases.

Figure 1 presents a family of $D_j(U_j)$ functions derived from Normal Probability Functions having an average bandwidth demand of 150 Mbit/s and different variance values σ^2 . For illustrative purposes the vertical line shows an assigned value of $U_j = 200$ Mbit/s. As can be seen, the assigned bandwidth of 200 Mbit/s satisfies demand (in terms of throughput) to various degrees, according to the variance of the demand function. For example, for $\sigma^2=0$, we observe a 100% satisfaction, whereas for $\sigma^2=150$ there is only about 80%. Expressed differently, for $\sigma^2=150$ to fully satisfy the demand the assigned bandwidth must be about 300 Mbit/s, whereas for $\sigma^2=25$ 150Mbit/s will suffice (i.e. for a mean demand of 150 Mbits/sec with a variance of 25, the assigned bandwidth of 200Mbit/s is overallocated).

Figure 1: Typical throughput functions $D_j(U_j)$.

Above problem belongs to the general class of Multiobjective non-linear constrained optimisation problems. We want to find the set of Pareto optimal solutions, and from this set select the optimum (or preferred) solution. This is defined as any preferred Pareto optimal solution that belongs to the indifference band (a subset of the Pareto optimal set where the improvement of one objective function is equivalent in

the mind of the decision maker – to the degradation of another [39]).

C. Solution approaches

Many methods that generate the set of feasible solutions of multiobjective BAVP [22], [40] exist. These include:

- The weighted sum (or convex combination) of the objective functions [31] (weighting method, parametric method). For example, Herzberg et al [29] converts the Multiobjective non-linear problem to a single objective LP problem, generating only one solution among the infinitely many.
- The ϵ -constrained method [24] which can produce the set of non-inferior solutions and (in conjunction with the Surrogate Worth Trade-off (SWT) [24] method) generate the relative trade-offs between the objective functions. Hence, it allows a quantitative comparison of the objective (even non-commensurate) functions (e.g. [10]).
- Hierarchical Multiobjective analysis that exploits the general concept of decomposition-co-ordination; it provides computational tractability, and possibly decentralisation of computations [24].
- Game theoretic concepts to deal with conflicting objectives, or introducing fairness into the VP allocations [37].
- Optimisation methods based on genetic algorithms, evolutionary strategies and evolutionary programming. Numerous suggestions for Multiobjective evolutionary computation algorithms exist [41, 42, 43], but note that theoretical results are still rather limited [44].

III. METHOD

In this study we investigate the reported strength of GA in conjunction with CCO to gain further insight into the problem of BAVP, focusing on single objective optimisation (using the sums of the objective functions of the individual VP's). Note that an initial investigation of the Multiobjective case was investigated in [10]. The formulations and solutions for the single objective of the sum and of the product of the objective functions were compared with the Multiobjective formulation and solution. It was shown that both sum and product formulations are subsets of the general Multiobjective solution. In the Multiobjective case, equipped with the Pareto optimum set, one can select the "best" solution, in the eyes of a decision maker (for example by using the Surrogate Worth Trade-off function method [38]).

A. Classical Constrained Optimisation method

The algorithm we use for the comparison is the Sequential Quadratic Programming (SQP) method as implemented in the Matlab Optimisation Toolbox [45]. In this method a Quadratic Programming (QP) subproblem is solved at each iteration. An estimate of the Hessian of the Lagrangian is updated at each iteration using the BFGS formula. A line search is performed using a merit function similar to that proposed by Hand and Powell [45, 46]. The QP subproblem

is solved using an active set strategy similar to that described in Gill, Murray, and Wright [47].

B. Genetic programming function optimisation method

Genetic Algorithms (GAs) [13] firstly introduced by John Holland in early seventies, are principal "search procedures" based on principles derived from dynamics of natural population genetics. They have been successfully applied to numerous large space problems where no efficient polynomial-time algorithm is known, such as NP-complete.

The problems to be solved in GAs should be represented as one-dimensional or multi-dimensional structure, which represent a search point in the search space. This means that the problem should be encoded, that is to find a pattern to represent each solution, called *chromosome*, as a chain of characters taken in a finite alphabet. The GAs operate on chromosomes grouped into a set called *population*. Successive populations are called *generations*. Each chromosome is evaluated by the *fitness function*, which reflects its merit and its chances to survive in the next generation.

In this study the GA algorithm for Numerical Optimisation for Constraint Problems (GENOCOP) based on the floating point representation is implemented as follows [13]:

step 1. A total of N real valued chromosomes, U_i , $i = 1, \dots, N$, ($P = N$) are initialized.

step 2. All chromosomes, U_i , $i = 1, \dots, N$, are evaluated with respect to the fitness function $F(U) =$

$$\max_U \left(\sum_i D_i(U_i) \right)$$

step 3. Some chromosomes of the population (the 'winners') reproduce, while others (the 'losers') die.

step 4. Genetic operators are applied on 'winners' and a new generation is produced to replace members that died. The genetic operators based on floating point representation of the GENOCOP system, as documented in [13], are briefly described in Appenix A.

step 5. During reproduction, randomly selected genetic operators are applied on random 'winner' chromosomes, one or two each time depending on operator, until all members that died are replaced.

Step 6. Go to *step 2* for a predetermined number of generations.

For a brief description of GENOCOP operators, see appendix A.

C. Hybrid Scheme

The third algorithm is the hybrid scheme. We investigate a combination of the above methods, which is expected to take advantage of the strength of each algorithm. We have implemented it as follows:

We first run GENOCOP and terminate it when a pre-set termination condition becomes true. We then switch over to CCO algorithm, using the results obtained by GENOCOP as input conditions. The termination condition we have set is $U_{g_n} - U_{g_{n-10}} < e$, where U stands for Utilisation, g_n is the n^{th} generation, g_{n-10} is the generation 10 steps ago, and e is set equal to 0.5.

IV. CASE STUDIES

We consider in detail two topologies for evaluating the optimisation methods: a 3-node and a 7-node network. We also solve 4-, 5- and 6-node problems, and provide some general observations regarding throughput, time complexity and fairness for all the cases considered. Also in the case of the GA algorithm we provide some observations regarding the selection of the tuning parameters.

A. CASE A: 3-node network, 2 OD pairs, 4 VPs

In [11], we studied a 3-node network ($N=3$) with 2 OD pairs, both destined for node 3. Two VP's were established for each OD pair ($N_p=4$, $P_j=2$, $j=1,2$) as shown in Table 1.

OD Pair	1	2
VP1 links	1-3	2-3
VP2 links	1-2-3	2-1-3

Table 1 : VP and OD pairs for the 3-node network (Case A); 4 VPs are set; 2 VPs for each of the 2 OD pair for the 3 node, 3 link network topology.

The link capacities were set equal to 100 Mbit/s. The network topology is shown in Figure 2, and the traffic characteristics (assuming a normally distributed probability function for bandwidth demand) are tabulated in Table 2.

Figure 2: Three-node network topology used for example 1.

Throughput and Convergence Comparison

For both cases (low and high variance), there was exact agreement between CCO-BAVP and GA-BAVP algorithms in the assignment of the bandwidth to the VPs. The convergence characteristics were also comparable. The convergence of the CCO-BAVP algorithm was within 50 iterations, whereas convergence for GA-BAVP algorithm was achieved within 5 generations. Both algorithms required similar computation time. (see Table 2).

B. CASE B. 7-node, 11-links, 42 OD pairs, 84 VPs

In topology B, (Figure 6) we study a 7-node network ($N=7$) with 11 links, 42 OD pairs and 84 defined VP's (Table 3).

Two VP's were established for each OD pair ($N_p=84$, $P_j=2$, $j=1,2$). Two cases were considered, with one case assigned scarce link capacities (set equal to 480 Mbits/sec) and the other set higher (and hence more relaxed) link capacities (720 Mbit/s). For the scarce link capacities, the OD pairs are shown in Table 3 and the traffic characteristics (assuming a normally distributed probability function for bandwidth demand) are tabulated in Table 6 (in Appendix B) shown together with the bandwidth assigned for each VP by the two algorithms.

Demand			Sum of objective functions, $\text{Max}\{D_1(U_1)+D_2(U_2)+\dots+D_{42}(U_{42})\}$			
OD pair number	Mean μ	Variance σ	CCO-BAVP		GA-BAVP	
			Bandwidth Allocations	Objective Function	Bandwidth Allocations	Objective Function
1	110	55	$U_1=137$	$D_1(U_1) = 99$	$U_1=137$	$D_1(U_1) = 99$
2	50	25	$U_2=63$	$D_2(U_2) = 45$	$U_2=63$	$D_2(U_2) = 45$
$\sum_i \mu_i = 160$			$\max_U \left(\sum_i D_i(U_i) \right) = 144$		$\max_U \left(\sum_i D_i(U_i) \right) = 144$	

Table 2: Traffic demands and bandwidth allocations for 3-node network (Case A) using CCO-BAVP in comparison to GA-BAVP optimisation techniques.

Figure 6: Seven-node network topology, showing only VP1 links because of the network complexity.

OD Pair	1	2	3	4	5	6	7	8	9
VP1 links	1-2	1-2-3	1-4	1-4-5	1-2-6	1-2-3-7	2-1	2-3	2-4
VP2 links	1-4-2	1-2-6-3	1-2-4	1-2-5	1-4-5-6	1-2-6-7	2-4-1	2-6-3	2-5-4

OD Pair	10	11	12	13	14	15	16	17	18
VP1 links	2-5	2-6	2-3-7	3-2-1	3-2	3-2-4	3-6-5	3-6	3-7
VP2 links	2-6-5	2-3-6	2-6-7	3-2-4-1	3-6-2	3-6-5-4	3-2-5	3-7-6	3-6-7

OD Pair	19	20	21	22	23	24	25	26	27
VP1 links	4-1	4-2	4-2-3	4-5	4-5-6	4-5-6-7	5-4-1	5-2	5-6-3
VP2 links	4-2-1	4-1-2	4-5-6-3	4-2-5	4-2-6	4-2-3-7	5-2-1	5-4-2	5-2-3

OD Pair	28	29	30	31	32	33	34	35	36
VP1 links	5-4	5-6	5-6-7	6-2-1	6-2	6-3	6-2-4	6-5	6-7
VP2 links	5-2-4	5-2-6	5-2-3-7	6-5-4-1	6-5-2	6-7-3	6-5-4	6-2-5	6-3-7

OD Pair	37	38	39	40	41	42
VP1 links	7-3-2-1	7-3-2	7-3	7-6-5-4	7-6-5	7-6
VP2 links	7-6-2-1	7-6-2	7-6-3	7-3-2-4	7-3-2-5	7-3-6

Table 3: VP and OD pairs for the 7-node network (Case B); 84 VPs are set; 2 VPs for each of the 42 OD pair for the 7 node, 11 link network topology.

Throughput and convergence comparison

For both CCO-BAVP and GA-BAVP algorithms, similar findings were obtained. However, the throughput computed by the classical method was slightly higher, by about 3%, than the one computed by the GA-BAVP method for the case of scarce bandwidth (see Table 6), and about 001% for the other case (see Table 4). Note that since the expected throughput function $D(U)$ peaks at the value of the mean demand function, then the maximum possible total throughput (for plentiful capacity that is not constraining the solution) is equal to the sum of the means of the demands; i.e. in this case it is equal to 3940. For the case of scarce bandwidth CCO

achieved a maximum of 2933 and GA 2840 (representing 74.5% and 72% respectively of the optimum total throughput of 3940). For the case of higher (less constraining) bandwidth, CCO achieved a maximum of 3528 and GA 3477 (representing 89.5% and 88.3% of the optimum total throughput of 3940 respectively), i.e. the difference between the two algorithms becomes much smaller as the capacity becomes plentiful.

As for the 4-node case, the GA-BAVP algorithm, for scarce and plentiful capacities, shows quick initial convergence to a value close to the optimal (within 5%), but with very slow progress afterward (see Figure 7). Although the initial convergence of the CCO-BAVP algorithm was

slow, it exhibited faster convergence towards the optimum, with a higher allocation than GA-BAVP, for a similar computational time.

Figure 7: Convergence for 7-node network topology problem (Case C) with CCO-BAVP (dotted line), GA-BAVP (continuous line) and Hybrid (dashed line) for the scarce link capacity case (480 Mbit/sec).

Fairness Comparison

We define “fairness” as the deviation of assigned capacity from capacity required to give 95% satisfaction of utility function $D_i(U_i)$ for each VP_i .

Figure 8: Deviation of assigned capacity for 7-node network topology problem (Case C) from the capacity required for 95% satisfaction of the utility function $D_i(U_i)$ for each VP_i . This graph represents a measure of fairness.

Figure 8 shows deviation of assigned capacity from capacity required to give 95% satisfaction of utility function $D_i(U_i)$ for each OD Pair i for both CCO-BAVP

and GA-BAVP (note if deviation is above 0 we have overallocation whereas below 0 we have underallocation).

Observe that GA-BAVP shows less deviation to the satisfaction of the 95% assignment than CCO-BAVP, for almost all OD Pairs. It is worth observing that CCO-BAVP allocates 0 bandwidth in two of the 42 OD Pairs (OD Pairs 6 and 37, see Table 6), and greatly overallocates capacity for two other OD Pairs (OD pairs 10 and 33) resulting in unfairness in the first instance and unnecessary waste of a significant amount of bandwidth in the second instance. Also worth noting is the variance of the deviation of assigned capacity from the capacity required to give 95% satisfaction of the utility function: for the CCO-BAVP technique it is equal to 2537 and for the GA-BAVP it is equal to 1881. Clearly, GA-BAVP is “fairer” in the sense of the fairness measure defined. Note that for the case of 720 Mbits/sec link capacity (more plentiful capacity) no obvious advantage in terms of “fairness” was observed by either algorithm.

C. General observations for all cases studied

Throughput

In Table 4 we present the results obtained by CCO-BAVP and GA-BAVP for a number of topologies ranging from 3 node to 7 node (2 OD pairs to 42 OD pairs) and for different link capacities. These are compared among themselves, as well as with the Optimum Utilisation obtained for infinite link capacities. As shown in Table 4, there is no clear winner with regard to the maximisation of the throughput. The two algorithms achieve comparable results (within a few % difference from each other), with some topologies and link capacities favoured by the CCO-BAVP and others by GA-BAVP. Also, as the link capacities are increased the utilisation achieved by both algorithms tends toward the optimum. For the case of 4-node topology we also confirmed this trend by increasing the link capacity by a 100 fold (i.e. 12000 Mbits/sec)—the utilisation achieved was within 1.4% from the Optimum Utilisation for both CCO-BAVP and GA-BAVP.

Topology	Bandwidth per link	Optimum Utilisation (for infinite link capacity)	CCO-BAVP		GA-BAVP	
			Utilisation	% deviation from optimum	Utilisation (number of generations)	% deviation from optimum
3-node (2 OD pairs)	100 Mbit/sec	170	139	18	139	18
4 node (12 OD pairs)	120 Mbit/sec	375	327	13	324	14
5 node (20 OD pairs)	270 Mbit/sec	1975	1429	28	1425 (2000) 1465 (10000)	28 26

5 node (20 OD pairs)	360 Mbit/sec	1975	1663	16	1642 (2000) 1674 (10000)	17 15
6 node (30 OD pairs)	360 Mbit/sec	2730	1950	29	1909 (2000) 1957 (10000)	30 28
6 node (30 OD pairs)	480 Mbit/sec	2730	2246	18	2249 (2000) 2283 (10000)	18 16
7 node (42 OD pairs)	480 Mbit/sec	3940	2933	26	2840 (10000)	28
7-node (42 OD pairs)	760 Mbit/sec	3940	3528	10	3477 (10000)	12

Table 4: Comparison of utilisation achieved for different network topologies for CCO-BAVP and GA-BAVP optimisation techniques

	CCO-BAVP		GA-BAVP	
	Solution time (seconds)	% increase of OD pair over the previous OD pair	Solution time seconds (number of generations)	% increase of OD pair over the previous OD pair
4 node, 12 OD pairs	127		1353 (2000)	
5 nodes, 20 OD pairs	593	467	3024 (2000) 14699 (10000)	223 (2000) 1086 (10000)
6 nodes, 30 OD pairs	1348	227	4761 (2000) 23879 (10000)	157 (2000) 162 (10000)
7 nodes, 42 OD pairs	2725	202	39671 (10000)	166 (10000)

Table 5: Comparison of solution times for different network topologies

Time Complexity

We investigated time complexity of CCO-BAVP and GA-BAVP algorithms for 3, 4, 5, 6 and 7 node network topologies. Ignoring the trivial solution of the 2 OD pairs, Table 5 compares the % increase (over the previous problem formulation solution) of CCO-BAVP and EP-BAVP.

It is clearly demonstrated that as the problem complexity increases, from 12 OD pairs to 42 OD pairs, the time complexity of the GA based algorithm does not increase as fast as the CCO solution. Note that a direct comparison is not feasible as the two algorithms are solved using different computers and languages (Matlab and C hand-code).

Hybrid Scheme

The experimental results have shown that GA-BAVP has an initial faster convergence than CCO-BAVP but CCO-BAVP exhibits slightly better solutions than GA-BAVP in the longer term (but less fair). Therefore, we implemented a hybrid scheme aiming to combine both methods, in order to take advantage of both of them and to check whether this combination gives better solutions. The experiments have shown that not only the solution we have obtained is not better than CCO-BAVP's solution, but also the capacity assignments per OD pair are exactly the same as the capacity assignments per OD pair for CCO-BAVP. The advantage this approach has is that it has an initial convergence the same as GA-BAVP and a solution as CCO-BAVP (see Figure 8), therefore it exhibits faster solution.

V. CONCLUDING REMARKS

In this paper we investigated the problem of bandwidth allocation using two different algorithms, the CCO-BAVP and the GA-BAVP with network topologies ranging from 3- to 7- node (2 to 42 OD pairs, 4 to 84 VPs). We compare their performance with regard to throughput maximisation, time complexity, and fairness:

- For the network topologies considered in this paper, both the CCO-BAVP and GA-BAVP algorithms give satisfactory solutions for throughput maximisation (see Table 4). The two algorithms achieve comparable results (within a few % difference from each other), with some topologies and link capacities favoured by the CCO-BAVP and others by GA-BAVP.
- As the problem complexity increases, from 12 OD pairs to 42 OD pairs, the time complexity of the GA based algorithm does not increase as fast as the CCO solution (see Table 5). Also, GA-BAVP outperforms CCO-BAVP in the initial convergence period, but after that CCO-BAVP exhibits faster convergence towards optimum. This suggested combining the two. The hybrid methodology exhibited better overall convergence rate but the same solution as CCO-BAVP.
- GA-BAVP exhibits "fairer" allocations of bandwidth, in the sense of the fairness measure defined earlier, than that of CCO-BAVP. It is worth observing that for the case of the 7-node topology (see Figure 8) the CCO-BAVP allocates 0 bandwidth in two of the 42 OD Pairs (Table 6), and greatly overallocates capacity for two other OD Pairs, resulting in unfairness in the first

instance and unnecessary waste of a significant amount of bandwidth in the second instance.

Based on the above results, the use of GA-BAVP can be justified on the basis of the 'fairer' bandwidths allocated to the competing Virtual Paths, the trends observed regarding the time solution for more complex topologies, and the achieved throughput.

Even though the proposed algorithm may be directly applicable to the allocation of bandwidths for small networks, such as Virtual Private Networks, it is highly desirable to extend the network topologies to more real life ones, and to include in future work a complete exploration of their properties, including fairness, implementation issues (including various speedups and parallelisation), as well extension to the Multiobjective optimisation bandwidth allocation problems. The hybrid methodology is another area which could be investigated further, as well as the integration of the bandwidth allocation problem with on line bandwidth prediction of demand patterns and dynamic network configuration.

Finally, the adaptation of these models to the Internet (e.g. in the differentiated services model for allocating aggregated bandwidth, in Virtual Private Networks, and networks using MPLS) is highly recommended.

Appendix A

The operators of the GENOCOP algorithm are briefly described here, as documented in [13]. The GENOCOP algorithm is concerned with optimizing a non-linear function $f(x_1, \dots, x_q) \in R$, where $(x_1, \dots, x_q) \in D \subseteq R^q$; D is a convex set, subject to a set of linear constraints. The domain D is defined by ranges of variables $l_k \leq x_k \leq r_k$ for $k=(1, \dots, q)$ and by a set of constraints. From the convexity of the set D it follows that for each point in the search space $(x_1, \dots, x_q) \in D$ there exists a feasible range $\langle left(k), right(k) \rangle$ of a variable x_k ($1 \leq k \leq q$), where other variables x_i ($i=1, \dots, k-1, k+1, \dots, q$) remain fixed (the convexity of the search space ensures that linear combinations of solutions yield solutions without needing to check the constraints).

The GENOCOP system tries to locate an initial (feasible) solution by sampling the feasible region. Once found (or given if initial search is not successful after a predefined number of steps), the initial population is set with identical copies of such an initial point. Then the GA algorithm is applied, as described in section II.D.

Six Genetic Operators are considered which guarantee to keep all chromosomes within the constrained solution space. The first three are unary operators (category of mutation) and the other three are binary operators (various types of crossovers).

Uniform mutation: This operator requires a single parent (chromosome) x and produces a single offspring x' . The operator selects a random component $k \in (1, \dots, q)$ of the

vector $x = (x_1, \dots, x_k, \dots, x_q)$ and produces $x' = (x_1, \dots, x'_k, \dots, x_q)$, where x'_k is a random value with uniform probability distribution in the range $\langle left(k), right(k) \rangle$.

Boundary mutation: This operator requires a single parent x and produces a single offspring x' . The operator is a variation of the previous operator, the uniform mutation, with x'_k being either $left(k)$ or $right(k)$, each with equal probability.

Non-Uniform mutation: This is a unary operator responsible for the fine-tuning capabilities of the algorithm. For a parent x , if the element x_k was selected for non-uniform mutation, the result is $x' = (x_1, \dots, x'_k, \dots, x_q)$, where

$$x'_k = \begin{cases} x_k + \Delta(t, right(k) - x_k) & \text{if a random binary digit is 0} \\ x_k - \Delta(t, x_k - left(k)) & \text{if a random binary digit is 1} \end{cases}$$

The function $\Delta(t, y)$ returns a value in the range $[0, y]$ such that the probability of $\Delta(t, y)$ being close to 0 increases as the number of generations t increases. This property causes this operator to search the space uniformly initially and very locally at later stages.

Arithmetic Crossover: This is a binary operator that is defined as a linear combination of two vectors: if x and y are to be crossed, the resulting offspring are $x' = a.x + (1-a).y$ and $y' = a.y + (1-a).x$. This operator uses a random value $a \in [0..1]$, as it always guarantees closure ($x', y' \in D$).

Simple crossovers: This binary operator is defined as follows: if $x = (x_1, \dots, x_q)$ and $y = (y_1, \dots, y_q)$ are crossed after the k -th position, the resulting offspring are: $x' = (x_1, \dots, x_k, y_{k+1}, \dots, y_q)$ and $y' = (y_1, \dots, y_k, x_{k+1}, \dots, x_q)$. It is noted that this operator may produce an offspring that is outside the domain D . To avoid this, the property of convex spaces is used, where there exists $a \in [0, 1]$ such that $x' = (x_1, \dots, x_k, y_{k+1}.a + x_{k+1}.(1-a), \dots, y_q.a + x_q.(1-a))$ and $y' = (y_1, \dots, y_k, x_{k+1}.a + y_{k+1}.(1-a), \dots, x_q.a + y_q.(1-a))$ are feasible.

Heuristic Crossover: this operator generates a single offspring z from the two parents x and y according to the following rule: $z = r.(y - x) + y$ where r is a random number between 0 and 1, and the parent y is not worse than x , i.e. $f(y) \geq f(x)$ for maximization problems and $f(y) \leq f(x)$ for minimization problems. It is possible for this operator to generate an offspring vector which is not feasible. In such a case another random value r is generated and another offspring is created.

Appendix B

Demand			Sum of objective functions, $\text{Max}\{D_1(U_1)+D_2(U_2)+\dots+D_{42}(U_{42})\}$			
OD pair number	Mean μ	Variance σ	CCO-BAVP		GA-BAVP	
			Bandwidth Allocations	Objective Function	Bandwidth Allocations	Objective Function
1	110	50	$U_1= 133.09$	$D_1(U_1)= 99.20$	$U_1= 143.47$	$D_1(U_1)= 102.03$
2	80	25	$U_2= 56.09$	$D_2(U_2)=53.82$	$U_2= 42.02$	$D_2(U_2)= 41.34$
3	120	35	$U_3= 183.01$	$D_3(U_3)=118.21$	$U_3= 185.94$	$D_3(U_3)= 118.28$
4	70	30	$U_4= 83.75$	$D_4(U_4)=63.54$	$U_4= 57.55$	$D_4(U_4)= 50.87$
5	95	40	$U_5= 58.03$	$D_5(U_5)=54.29$	$U_5= 35.28$	$D_5(U_5)= 34.21$
6	115	20	$U_6= 0$	$D_6(U_6)=0$	$U_6= 43.29$	$D_6(U_6)= 43.29$
7	120	40	$U_7= 138.44$	$D_7(U_7)=111.01$	$U_7= 148.00$	$D_7(U_7)= 113.56$
8	80	45	$U_8= 76.96$	$D_8(U_8)=61.04$	$U_8= 59.62$	$D_8(U_8)= 50.70$
9	100	30	$U_9= 115.86$	$D_9(U_9)=93.86$	$U_9= 118.45$	$D_9(U_9)= 94.56$
10	130	50	$U_{10}= 237.75$	$D_{10}(U_{10})=127.95$	$U_{10}= 164.31$	$D_{10}(U_{10})= 121.90$
11	100	65	$U_{11}= 97.02$	$D_{11}(U_{11})=74.09$	$U_{11}= 129.81$	$D_{11}(U_{11})= 87.55$
12	85	16	$U_{12}= 75.84$	$D_{12}(U_{12})=72.94$	$U_{12}= 61.42$	$D_{12}(U_{12})= 60.91$
13	100	70	$U_{13}= 34.93$	$D_{13}(U_{13})=30.70$	$U_{13}= 19.14$	$D_{13}(U_{13})= 17.26$
14	50	55	$U_{14}= 46.39$	$D_{14}(U_{14})=31.65$	$U_{14}= 47.59$	$D_{14}(U_{14})= 32.27$
15	110	35	$U_{15}= 79.34$	$D_{15}(U_{15})=75.62$	$U_{15}= 41.91$	$D_{15}(U_{15})= 41.58$
16	80	60	$U_{16}= 75.87$	$D_{16}(U_{16})=56.39$	$U_{16}= 40.84$	$D_{16}(U_{16})= 34.12$
17	55	40	$U_{17}= 143.93$	$D_{17}(U_{17})=55.52$	$U_{17}= 86.40$	$D_{17}(U_{17})= 51.36$
18	145	20	$U_{18}= 161.19$	$D_{18}(U_{18})=141.74$	$U_{18}= 156.09$	$D_{18}(U_{18})= 140.60$
19	60	40	$U_{19}= 134.65$	$D_{19}(U_{19})=59.93$	$U_{19}= 132.43$	$D_{19}(U_{19})= 59.88$
20	90	45	$U_{20}= 113.47$	$D_{20}(U_{20})=81.34$	$U_{20}= 149.83$	$D_{20}(U_{20})= 87.57$
21	100	30	$U_{21}= 73.42$	$D_{21}(U_{21})=70.29$	$U_{21}= 53.84$	$D_{21}(U_{21})= 53.04$
22	100	50	$U_{22}= 126.21$	$D_{22}(U_{22})=90.38$	$U_{22}= 130.66$	$D_{22}(U_{22})= 91.60$
23	130	65	$U_{23}= 74.91$	$D_{23}(U_{23})=68.24$	$U_{23}= 76.39$	$D_{23}(U_{23})= 69.41$
24	115	10	$U_{24}= 31.83$	$D_{24}(U_{24})=31.82$	$U_{24}= 8.20$	$D_{24}(U_{24})= 8.20$
25	60	55	$U_{25}= 85.91$	$D_{25}(U_{25})=52.27$	$U_{25}= 49.91$	$D_{25}(U_{25})= 36.49$
26	70	40	$U_{26}= 154.87$	$D_{26}(U_{26})=69.41$	$U_{26}= 99.68$	$D_{26}(U_{26})= 64.96$
27	50	50	$U_{27}= 46.96$	$D_{27}(U_{27})=32.67$	$U_{27}= 45.88$	$D_{27}(U_{27})= 32.10$
28	120	10	$U_{28}= 124.82$	$D_{28}(U_{28})=117.43$	$U_{28}= 133.09$	$D_{28}(U_{28})= 118.79$
29	90	20	$U_{29}= 88.25$	$D_{29}(U_{29})=80.93$	$U_{29}= 93.96$	$D_{29}(U_{29})= 83.59$
30	100	60	$U_{30}= 67.33$	$D_{30}(U_{30})=57.41$	$U_{30}= 57.83$	$D_{30}(U_{30})= 50.47$
31	80	30	$U_{31}= 52.07$	$D_{31}(U_{31})=49.25$	$U_{31}= 83.48$	$D_{31}(U_{31})= 69.54$
32	130	65	$U_{32}= 126.30$	$D_{32}(U_{32})=102.38$	$U_{32}= 153.34$	$D_{32}(U_{32})= 113.99$
33	50	60	$U_{33}= 184.99$	$D_{33}(U_{33})=55.49$	$U_{33}= 97.72$	$D_{33}(U_{33})= 49.25$
34	70	30	$U_{34}= 44.49$	$D_{34}(U_{34})=41.30$	$U_{34}= 50.79$	$D_{34}(U_{34})= 46.15$
35	130	45	$U_{35}= 127.01$	$D_{35}(U_{35})=110.17$	$U_{35}= 131.47$	$D_{35}(U_{35})= 112.38$
36	115	60	$U_{36}= 163.95$	$D_{36}(U_{36})=107.75$	$U_{36}= 132.25$	$D_{36}(U_{36})= 98.86$
37	50	10	$U_{37}= 0$	$D_{37}(U_{37})=0$	$U_{37}= 18.82$	$D_{37}(U_{37})= 18.82$
38	30	25	$U_{38}= 16.43$	$D_{38}(U_{38})=13.26$	$U_{38}= 27.70$	$D_{38}(U_{38})= 20.31$
39	150	80	$U_{39}= 215.59$	$D_{39}(U_{39})=140.39$	$U_{39}= 165.03$	$D_{39}(U_{39})= 125.35$
40	90	30	$U_{40}= 13.30$	$D_{40}(U_{40})=13.26$	$U_{40}= 30.07$	$D_{40}(U_{40})= 29.83$
41	110	60	$U_{41}= 77.19$	$D_{41}(U_{41})=66.88$	$U_{41}= 72.18$	$D_{41}(U_{41})= 63.28$
42	105	40	$U_{42}= 137.34$	$D_{42}(U_{42})=99.62$	$U_{42}= 139.24$	$D_{42}(U_{42})= 99.99$
$\sum_i \mu_i = 3940$			$\max_U \left(\sum_i D_i(U_i) \right) = 2933$		$\max_U \left(\sum_i D_i(U_i) \right) = 2840$	

Table 6: Traffic demands and bandwidth allocations for 7-node network (Case B) using CCO-BAVP in comparison to GA-BAVP optimisation techniques for the case of scarce link bandwidth (480 Mbit/s)

REFERENCES

- [1] S. Ohta and K.-I. Sato, Dynamic bandwidth control of the virtual path in an asynchronous transfer mode network, *IEEE Transactions in Communication*, 40, 7, 1992. pp.1239-1247
- [2] R.G. Addie, J.L. Burgin, S.L. Sutherland, B-ISDN protocol architecture, *IEEE Globecom'88*, Florida, USA, December 1988.
- [3] D. Clark, J. Wroclawski, An approach to service allocation in the Internet, Internet draft, July 1997.
- [4] L. Zheng, S. Deering, S. Estrin, S. Shenker, D. Zappala, A new Resource Reservation protocol, *IEEE Network*, September 1993.
- [5] Z. Wang, User-share differentiation-scalable service allocation for the internet, Internet draft, November 1997.
- [6] Z. Wang, Towards scalable bandwidth allocation in the Internet, <http://www.bell-labs.com/user/zhwang/diff-serv/article.txt>.
- [7] D. Awduche, "MPLS and Traffic Engineering in IP Networks," *IEEE Communications Magazine*, vol. 37, no. 12, Dec. 1999, pp. 42-47.
- [8] G. Swallow, "MPLS Advantages for Traffic Engineering," *IEEE Communications Magazine*, vol. 37, no. 12, Dec. 1999, pp. 54-57.
- [9] J. Lawrence, Designing Multiprotocol Label Switching Networks, *IEEE Communications Magazine*, vol. 39, no. 7, July 2001, pp. 134-142.
- [10] A. Pitsillides, "Control structures and techniques for Broadband-ISDN communication systems", Ph.D. Thesis, Swinburne University of Technology, 1993.
- [11] A. Pitsillides, C. Pattichis, A. Sekercioglu, A. Vasilakos, "Bandwidth Allocation for Virtual Paths using Evolutionary Programming (EP-BAVP)", *ICT'97, International Conference on Telecommunications*, Melbourne, Australia, 2-4 April 1997.
- [12] A. Pitsillides, C. Pattichis, G. Stylianou, A. Sekercioglu, A. Vasilakos, "Investigation of the performance of EP-BAVP (Evolutionary Programming for Virtual Path Bandwidth Allocation)", *ICT'98, International Conference on Telecommunications*, Halkidiki, Greece, 22-24 June 1998.
- [13] Z. Michalewicz, "Genetic Algorithms + Data Structures = Evolution Programs", Third edition, Springer, Berlin, Germany, 1995.
- [14] D.B. Fogel, "Evolutionary Computation", IEEE Press, NJ, USA, 1995.
- [15] W. Pedrycz, A. V. Vasilakos (Ed.), *Computational Intelligence in Telecommunications Networks*, CRC Press, ISBN: 0-8493-1075-X, September 2000.
- [16] On-line bibliography on the application of genetic algorithms to the design and optimisation of data networks (1994-1998), <http://www.ee.cornell.edu/~bjhaskar/ganet-bib.html> (>90 papers listed).
- [17] H. Pan, I.Y. Wang, The bandwidth allocation of ATM through Genetic Algorithm, *Globecom'91*, vol. 1, pp. 125-129, 1991.
- [18] N. Shimamoto, A. Hiramatsu, K. Yamasaki, A dynamic routing control based on a Genetic Algorithm, *IJCNN'93*, pp. 1123-1128, 1993.
- [19] S. Tanterdtid, W. Steanputtangaul, W. Benjapolakul, Optimum virtual paths system based in ATM network using genetic algorithm, *Information, Communications and Signal Processing, ICICS'97*, pp. 596-601, vol.2, Singapore, September, 1997.
- [20] S. Tanterdtid, W. Steanputtangaul, W. Benjapolakul, Optimizing ATM network throughput based on virtual path concept by using genetic algorithm *IEEE Intelligent Processing Systems*, 1997. *ICIPS '97*, pp. 1634 -1639 vol.2, October 28-31, Beijing, China, 1997.
- [21] N. Swaminathan, J. Srinivasan, S.V. Raghavan, Bandwidth-demand prediction in virtual path in ATM networks using genetic algorithms, *Journal of Computer Communications*, 22, pp. 1127-1135, 1999.
- [22] M. Markaki, A. Vasilakos, I. Kassotakis, "A hybrid genetic algorithm for an effective management of isochronous channels in high speed networks", *IEEE International conference on Evolutionary Computation (ICEC'97)*, Purdue university, April 1997.
- [23] I.Kassotakis, M.Markaki, A.Vasilakos, "A Hybrid Genetic Approach for Channel Reuse in Multiple Access Telecommunication Networks", *IEEE Journal on Selected Areas in Communication*, Special Issue: Computational Intelligence in High Speed Networks (to appear 1st quarter 2000).
- [24] R. Elbaum, M. Sidi, "Topological design of Local Area Networks using Generic Algorithms", *IEEE/ACM Transactions of Networking*, Vol. 4, No. 5, October 1996.
- [25] K.T. Ko, K.S. Tang, C.Y. Chan, K.F. Man and S. Kwong, "Using Genetic Algorithms to Design Mesh Network", *IEEE Computer*, vol. 30, no. 8, pp. 56-61, 1997
- [26] M. Gerla, J. A. S. Monteiro, R. Pazos, "Topology Design and Bandwidth Allocation in ATM nets", *IEEE JSAC*, Vol. 7, No. 8, pp. 1253-1262, Oct. 1989.
- [27] J. Y. Hui, M. B. Gursoy, N. Moayeri, R. D. Yates, "A layered broadband switching architecture with physical or virtual path configurations", *IEEE Journal of Selected Areas in Communications*, Vol. 9, No. 9, pp. 1416-1426, December 1991.
- [28] M. Herzberg, "A linear programming model for virtual path allocation and management in B-ISDN", *ABSSS'92*, Melbourne, July 1992.
- [29] M. Herzberg, A. Pitsillides, "A hierarchical approach for the bandwidth allocation, management and control in B-ISDN", *IEEE International Conference on Communications, ICC'93*, Geneva, May 1993.
- [30] U. Mocci, P. Pannunzi and C. Scoglio, Adaptive capacity management of Virtual Path networks, *IEEE Globecom'96*, London, November 1996.
- [31] L. A. Zadeh, "Optimality and non-scalar-valued performance criteria", *IEEE Aut. Control*, AC-8, 1963.
- [32] A. Pitsillides, J. Lambert, D. Tipper, "An illustrative hierarchical structure for the allocation of bandwidth to virtual paths", *ATNAC94*, Australian Telecommunication Networks and Applications Conference, Melbourne, 5-7 December 1994.
- [33] E.W.M. Wong, A.K.M. Chan, S. Chan and K.T. Ko, "Bandwidth Allocation and Dynamic Routing in Virtual

-
- Path Based ATM Networks," Proceedings, Globecom '96, London, November 1996.
- [34] R. Bolla, F. Davoli, M. Marchese, "Bandwidth allocation and admission control in ATM networks with service separation", IEEE Communications Magazine, vol. 35, no. 5, pp. 130-137, May 1997.
 - [35] A.A. Ionomides, J.A. Silvester, Multiobjective routing in integrated services networks: a game theory approach, Infocom'91, 1991, pp. 1220-1225.
 - [36] R. Cocci, D. Estrin, S. Shenker, L. Zhang, Pricing in computer networks: motivation, formulation and example, IEEE/ACM Transactions of Networking, vol.1, February 1993, pp. 614-627.
 - [37] A. Lazar, A. Orda, D.E. Pendarakis, Virtual Path bandwidth allocation in multiuser networks, IEEE/ACM Transactions of Networking, Vol. 5, No. 6, December 1997, pp. 861-871.
 - [38] J. Von Neumann, O. Morgenstern, "Theory of games and economic behaviour", Princeton Uni., 1953.
 - [39] Y. Y. Haimes, K. Tarvainen, T. Shima, J. Thadathil, "Hierarchical Multiobjective Analysis of Large-Scale Systems", Hemisphere Publishing, 1990.
 - [40] J. G. Lin, "Three methods for determining Pareto-optimal solutions of multiple-objective problems", Directions in Large-Scale Systems, Y-C. Ho, S. K. Mitter (editors), Plenum Press, 1975.
 - [41] C. M. Fonseca, J.P. Fleming, "An overview of evolutionary algorithms in multiobjective optimisation", Evolutionary Computation, 3(1), pp. 1-16, 1995.
 - [42] H. Tamaki, H. Kita, S. Kobayashi, "Multiobjective optimisation by genetic algorithms: a review", ICEC'96, IEEE International Conference in Evolutionary Computation, Piscataway, NJ, pp. 517-522, 1996.
 - [43] J. Horn, "Multicriterion Decision Making", In T. Back, D.B. Fogel, Z. Michalewicz, (ed.) Handbook of Evolutionary Computation, Publishing and Oxford University Press, pp. F1.9:1-15, 1997.
 - [44] G. Rudolph, "On a multiobjective evolutionary algorithm and its convergence to the Pareto Set", ICEC'98, IEEE International Conference on Evolutionary Computation, Anchorage, Alaska, pp. 511-516, 4-9 May 1998.
 - [45] Matlab Optimisation Toolbox 2.0, Mathworks Inc.
 - [46] M. J. D. Powell. A fast algorithm for nonlinearly constrained optimization calculation. In G.A. Watson, editor, Numerical Analysis Proceedings, Dundee, 1978. Springer-Verlag.
 - [47] Gill, P.E., W. Murray and M.H. Wright, *Practical Optimization*, Academic Press, New York, 1981.