Developments in Theoretical and Applied Mechanics, Vol. XVIII, eds. H. Wilson, R. Batra, C. Bert, A. Davis
R. Schapery, D. Stewart, and F. Swinson, School of Engineering, The University of Alabama, 1996, pp. 411-424.

GENETIC ALGORITHMS AND OPTIMIZING CHEMICAL OXYGEN-IODINE LASERS

David L. Carroll[1]

ABSTRACT

This paper presents results from the first known application of the genetic algorithm (GA) technique for optimizing the performance of a laser system (chemical, solid-state, or gaseous). The effects of elitism, single point and uniform crossover, creep mutation, different random number seeds, population size, niching and the number of children per pair of parents on the performance of the GA for this problem were studied. Micro-GAs (μGA) were also tested. The best overall performer was the uniform crossover μGA with a population size of 5. The uniform crossover μGA was also able to find the global maximum of an order-3 deceptive function which the other tested GAs failed to optimize.

Nomenclature

icreep	= creep mutation flag (0 = false, 1 = true)
idum	= initial random number seed
ielite	= elitism flag (0 = false, 1 = true)
iniche	= niching flag (0 = false, 1 = true)
iuniform	= crossover flag (0 = single-point, 1 = uniform)
n_c	= number of chromosomes (bits) in individual's binary string
nchild	= number of children per pair of parents
n_p	= number of parameters in individual's binary string
n_{pop}	= population size
p_{creep}	= probability of creep mutation per parameter
p_{cross}	= probability of crossover
p_{jump}	= probability of jump mutation per chromosome (bit)
x_{i}	= Molar flow rate of species i (moles/s)
pri	= primary flow
sec	= secondary flow

[1] Research Scientist, University of Illinois, Urbana, Illinois 61801

I. Introduction

A chemical laser is a device which uses a series of chemical reactions between fluid dynamically mixing gases to obtain excited atoms (or molecules) for subsequent lasing. The typical chemical oxygen-iodine laser (COIL) utilizes an energy transfer from the singlet delta excited state of oxygen $O_2(^1\Delta)$ directly and indirectly to I_2 to dissociate the iodine molecule. This process is followed by an energy transfer from other $O_2(^1\Delta)$ molecules to the liberated iodine atoms, thus providing the energy for the atomic iodine laser transition of interest. A number of papers have investigated issues associated with the operation of chemical oxygen-iodine laser (COIL) systems. While there are numerous COIL related papers, a good summary of COIL technological development is provided by Truesdell (1992). However, only recently have researchers studied the effects of high pressure operation on small signal gain, iodine dissociation, the extent of mixing, power and system performance. The motivation for such research is that, if the total pressure of a COIL device can be increased significantly while maintaining the device's nominal performance, then it may be possible to reduce the size of pressure recovery systems. Carroll (1995a) addressed the issues of the small signal gain, power, and system performance when the total pressure of the flow is increased significantly. Barmashenko (1994) addressed similar issues at much lower operating pressures. Tate (1994) has performed detailed two-dimensional gain measurements of a COIL flow field as the total pressure of the flow was increased up to around 130 Torr. Helms (1994) recently investigated the effects of total pressure on the iodine dissociation up to 100 Torr and Scott (1994) has discussed the effects of total pressure on penetration and mixing characteristics. Madden (1994) and Buggeln (1994) have performed detailed computational fluid dynamic (CFD) mixing calculations for COIL flow fields having total pressures of around 70 Torr and 40 Torr, respectively.

To better understand how to optimize the performance of a high pressure COIL device, the effects of increased total pressure on mixing, kinetics, gain and power must be determined. To address these issues, the Blaze II computer model (Sentman, 1977) was used extensively to predict the performance of the research assessment and device improvement chemical laser (RADICL) COIL device.(Carroll, 1995a) With the aid of a genetic algorithm (GA) search and optimization technique, for the purposes of finding a set of unknown parameters (in a phase space having approximately 2 million permutations), this modeling effort produced very good agreement with power data as a function of the diluent ratio [$\psi = \dot{x}_{(\mathrm{He})_{\mathrm{pri}}}/\dot{x}_{(\mathrm{Cl}_2)_0}$] and the titration ratio [$\beta = \dot{x}_{\mathrm{I}_2}/\dot{x}_{\mathrm{O}_2} = \dot{x}_{\mathrm{I}_2}/U\dot{x}_{(\mathrm{Cl}_2)_0}$ where $U = \dot{x}_{\mathrm{O}_2}/\dot{x}_{(\mathrm{Cl}_2)_0}$ = the utilization of the Cl_2 flow rate into the oxygen generator].(Carroll, 1995a) This modeling provided important insights about the interactions between flow rates, kinetics, mixing, and pressure on COIL performance. The next step in the RADICL optimization process was to expand the search to higher pressure operating conditions using different geometries with the goal of using qualitative predictions to help guide future experiments.

II. The Blaze II Model

The Blaze II code (Sentman, 1977) was originally written to be as generic a chemical laser model as possible. Blaze II can treat arbitrary combinations of chemical species characterized by an adequate number of reactions (as many as 500 reactions and 40 species). Blaze II, which contains one-dimensional fluid dynamic equations whose mixing terms were derived from the two-dimensional equations that describe the mixing flow field in a chemical laser cavity, can be used for axisymmetric and 2D flows. Lasing may occur on a single atomic transition or on as many as 10 vibrational bands of a rotational equilibrium flow. Computer solution time is sufficiently short that extensive parametric studies can be performed in reasonably short times; a typical Hydrogen-Fluoride (HF) chemical laser run takes approximately 70-100 CPU seconds on an IBM RS/6000 computer and a typical COIL run takes approximately 30-60 CPU seconds on the same machine. Carroll (1995a) used the Blaze II code extensively to predict the performance of a COIL (single atomic transition) device. Since high pressure COIL experiments have been performed with the RADICL nozzle at Phillips Laboratory, these modeling investigations focused on that nozzle. A top view schematic of the RADICL device is shown in Fig. 1. A side view schematic of the RADICL nozzle and laser cavity section are shown in Fig. 2. Details of the RADICL device are given by Crowell (1993) and Carroll (1995a).

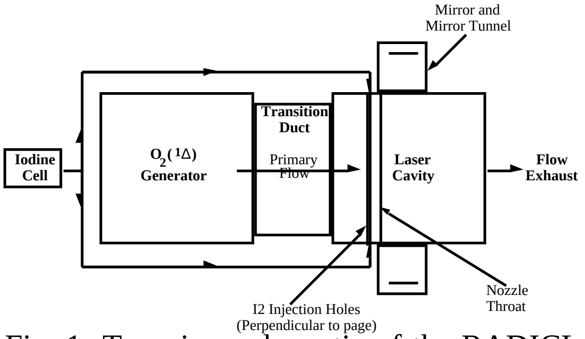

Fig. 1 Top view schematic of the RADICL experimental layout.

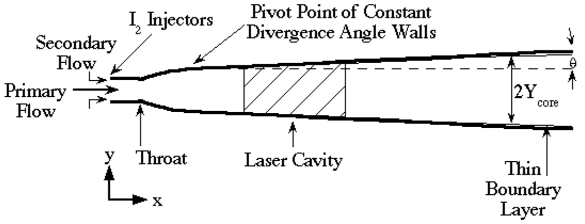

Fig. 2 Side view schematic of the RADICL nozzle showing nominal wall divergence angle, injector, throat, and laser cavity locations.

The previous COIL modeling of the RADICL device was performed by matching power calculations from a Fabry-Perot model (Blaze II) to stable resonator power data.(Carroll, 1995) While this is an acceptable approach when no other data (such as gain data and/or Fabry-Perot data) are available, it is generally believed that matching a model to gain data is a better method because the complicated mode-media interaction effects of lasing do not have to be modeled (a difficult and computationally expensive task if physical optics are included). With the recent availability of gain data for the RADICL device,(Tate, 1994) the Blaze II model was baselined to this gain data and predicted Fabry-Perot powers were compared with stable resonator data.(Carroll, 1995b) For the five cases which were matched to gain data, the predicted power was an average of approximately 33% higher than the actual measured power (or ≈4/3 times the measured power). Conversely, actual measured stable resonator powers were an average of 75% (≈3/4) of the predicted Fabry-Perot powers. Thus, any predicted powers discussed later are assumed to be 33% too large, and that likely measured powers for flow conditions which have not been experimentally tested will be assumed to be 75% of the predicted values. Based on a comparison of the predicted powers with measured powers, the power predictions of Blaze II must be considered qualitatitive in nature (Carroll, 1995b).

III. Optimizing High Pressure Performance

With the Blaze II model baselined to RADICL gain data, a genetic algorithm (GA) (Holland, 1975; Goldberg, 1989a) was implemented to optimize laser performance as a function of the flow rates [$\dot{x}_{(Cl_2)_0}$, $\dot{x}_{(He)_{pri}}$, $\dot{x}_{I_2}$, $\dot{x}_{(He)_{sec}}$], mirror position and nozzle configuration. The four flow rates represent four parameters, the mirror location another, and there are three relevant nozzle configuration parameters that can be readily adjusted in the RADICL device. The adjustable nozzle geometry factors are the addition of a spacer plate between the injectors and the throat (increasing the subsonic section of the flow), the throat height (which was chosen as 0.353", 0.56", 0.65", or 0.75"), and the laser cavity expansion angle which can be varied from 0° to 5° (these ramps were set to 3° for nominal operating conditions), Fig. 2. This is the first known application of the genetic algorithm technique for optimizing the performance of a laser system (chemical, solid-state, or gaseous).

One of the objectives of the GA optimization search was to determine if there were any systematic trends associated with the parameters, e.g., a priori, it is anticipated that the highest powers will always be associated with the highest allowable chlorine flow rates $\dot{x}_{(Cl_2)_0}$ [higher initial chlorine flow results in more oxygen flow and consequently more excited oxygen]. The 8 parameters were assigned realistic ranges for high pressure operating conditions, Table 1. The maximum values of the gas flow parameters are based on estimates of the current limitations of the RADICL device.(Helms, 1995) The GA search space was restricted to high pressure conditions having a primary He flow rate of 4.5-7.5 moles/s and an iodine flow rate in the range of 12.0-19.0 mmoles/s, Table 1. The lower limit of 4.5 moles/s of primary He was chosen to keep the total pressure above 100 Torr. The search space did not include iodine flow rates less than 12.0 mmoles/s because Helms (1995) experienced power instabilities with RADICL at titration ratios less than 0.010 (12 mmoles/s of I_2 combined with a Cl_2 flow rate of 1.3 moles/s corresponds to β=0.011). The 8 parameters were descritized and then translated into a binary string of length 22, Table 1. There are approximately 4 million (=2^{22}) possible permutations of this parameter space. The successful use of the GA technique for searching large chemical laser parameter spaces was demonstrated for the first time by Carroll (1995a).

Table 1. Genetic algorithm parameter search space for Blaze II.

Parameter	Range	Increment	# of Possibilities	# of Binary Digits
$\dot{x}_{(Cl_2)_0}$ [moles/s]	1.0 - 1.3	0.1	4	2
$\dot{x}_{(He)_{pri}}$ [moles/s]	4.5 - 8.0	0.5	8	3
$\dot{x}_{I_2}$ [mmoles/s]	12.0 - 19.5	1.0	8	3
$\dot{x}_{(He)_{sec}}$ [moles/s]	0.75 - 2.25	0.10	16	4
Mirror Leading Edge [cm]	5.2 - 12.2	1.0	8	3
Spacer Plate Size [in]	0.0, 0.25, 0.50, 0.75	---	4	2
Throat Height [in]	0.353, 0.56, 0.65, 0.75	---	4	2
Cavity Divergence Angle [°]	1.5 - 5.0	0.5	8	3

Goldberg's (1989a) simple GA used binary coding, roulette wheel (stochastic) selection and jump mutations. The basic GA of this study also uses jump mutations, but tournament selection is used because it is generally considered an improved selection technique (Goldberg, 1991a). Unless otherwise noted in the text or figures, the nominal GA (hereafter referred to as the "loaded" GA) of this study uses binary coding, tournament selection, uniform crossover, jump and creep mutations, niching, elitism and one child per pair of parents. Based on previous successful experience with GAs,(Carroll, 1996) the probability of crossover p_{cross} was set to 0.5 for both single-point and uniform crossover, the jump mutation probability p_{jump} was set equal to $1/n_{pop}$, and the creep mutation probability p_{creep} was set equal to $2/n_{pop}$ for all of the GA runs in this study.

The issue of what the creep probability should be set to has not previously been addressed in genetic algorithm work. For this problem, the use of $p_{jump} = 1/n_{pop}$ and $p_{creep} = 2/n_{pop}$ approximately matched the number of jump and creep mutations in a given population size. A more precise analysis can be made using basic probabilistic arguments. The overall probability of a jump mutation occuring for an individual i is

$$\left(p_{jump}\right)_i = 1 - \left(1 - p_{jump}\right)^{n_c} \tag{1}$$

where n_c is the number of chromosomes (bits) in an individual's binary string. The overall probability of a creep mutation occuring for an individual i is

$$\left(p_{creep}\right)_i = 1 - \left(1 - p_{creep}\right)^{n_p} \tag{2}$$

where n_p is the number of parameters in an individual's binary string. If we want the overall probability of jump and creep mutations to be the same, then Eqs. (1) and (2) can be set equal to each other and solved for p_{creep} giving

$$p_{creep} = 1 - \left(1 - p_{jump}\right)^{n_c / n_p} \tag{3}$$

Taking a binomial expansion of Eq. (3) and neglecting lower order terms since $p_{jump} << 1$, gives

$$p_{creep} \quad \frac{n_c}{n_p} p_{jump} = \frac{n_c}{n_p} \left(\frac{1}{n_{pop}} \right) \tag{4}$$

For this application where n_c=22 and n_p=8, Eq. (4) indicates that p_{creep} should be equal to 2.75/n_{pop}; this is roughly the same as the 2/n_{pop} probability which was used in this study.

The Blaze II model was automated for many calculations based on the eight parameters and the genetic algorithm technique was coded and implemented to search the parameter space and maximize the predicted power at high pressure conditions. Each function evaluation consists of one complete Blaze II calculation; each COIL run using Blaze II takes around 30-60 CPU sec on an IBM RS/6000 computer, therefore each function evaluation takes 30-60 CPU sec on this machine. Approximately 1.5 days of continuous running on a RS/6000 was required to run 26 generations for a population size of 100; 3 days for a population size of 200; 0.75 days for a population size of 50. This is the second known application of the genetic algorithm technique for modeling lasers, chemically reacting flows and chemical lasers.

The question of what is an appropriate population size must be addressed before any GA calculations can be run. From Goldberg (1992), an appropriate population size is

$$n_{pop} = O(m\, \chi^k) = O[(l/k)\, \chi^k] \tag{5}$$

where $m=l/k$, χ is the cardinality of chromosomes (the number of possibilities for each chromosome, e.g. for binary χ=2), k is the size of the schema (Holland, 1975; Goldberg, 1989a) of interest, and l is the length of the chromosome string. For the purposes of estimating an appropriate population size, it was assumed that each parameter string would represent one important schema; therefore, the schema length was assumed to be equal to the average parameter length for estimation purposes. For binary coding, l =22, χ=2 and the average length of the schema of interest is $k \approx (1/8)(2+3+3+4+3+2+2+3) \approx 3$ (this is the average length of the number of chromosomes which make up one parameter), so the population size should be on the order of $(22/3)(2^3) \approx 60$. Thus, population sizes of 50, 100 and 200 were tried for this problem.

The optimization search with the loaded-GA showed the anticipated trend of maximizing the Cl_2 (or equivalently the O_2) flow rate, along with minimizing the He_{pri} flow rate (which lowers the pressure and He deactivation in the flow, while increasing the diffusional mixing rate). For high Cl_2 flow rate conditions, minimizing the titration ratio to values of $\beta \approx 0.010$ maximized the power (discussed below). The search in the high pressure flow regime (100-250 Torr) showed a preference for maximizing the penetration by maximizing the secondary He flow rate. Optimal powers under high pressure conditions tended to occur with the mirror leading edge location further downstream; this may reflect the fact that there is more mixed flow further downstream. The search indicated that the addition of a spacer plate tended to decrease the laser performance; this is an indication that too much deactivation occurs in the subsonic region when a spacer plate is added and suggests that it may be beneficial to reduce the spacing between the injectors and the throat. The larger throat heights of 0.65" and 0.75" were preferred; this trend is not surprising because the plenum pressures are lower in these cases (resulting in faster diffusive mixing) and the high flow rate conditions lead to higher velocities which lower the $O_2(^1\Delta)$ transport losses. The larger expansion angles of 4.5° and 5.0° were preferred; this is believed to be a consequence of the resulting lower temperatures in the flow. However, there was typically only a 1-2% improvement in power when the expansion angle was increased from 3.0° to 5.0°. It is important to point out that conclusions based on these trends may only be appropriate for the flow regime studied in Table 1.

The GA found an optimal set of the parameters with a maximum predicted power of 26.5 kW; since the predicted powers using the baseline to gain data were approximately 33% higher than were measured powers (discussed above), the likely measured power should be roughly 19.9 kW. The parameter set that provided the optimal power had flow rates of 1.3 moles/s of Cl_2, 4.5 moles/s of He_{pri}, 12.0 mmoles/s of I_2, 2.25 moles/s of He_{sec}, the mirror leading edge at 12.2 cm, no spacer plate, a throat height of 0.75", and a 4.5° angle of divergence in the cavity (a near-optimal power was found with a 5.0° divergence angle). The total pressure of this flow was approximately 100 Torr. The diluent ratio for this case is ψ=3.5 and the titration ratio is β=0.011. As discussed above, the present search did not include iodine flow rates less than 12.0 mmoles/s because Helms (Helms, 1995) experienced power instabilities with RADICL at titration ratios less than 0.010. Most importantly, this modeling indicates that it should be possible to obtain significant power levels (> 19 kW) with the RADICL device under high pressure operating conditions.

IV. Effects of Different GA Techniques

A roll of the dice. Of interest is the effect of different random number seeds on the performance of the GA. Using the nominal loaded-GA discussed earlier with a population size (n_{pop}) of 100, two different initial random number seeds (*idum*) were tested, Fig. 3. With *idum* set to -10000, the GA found the optimal power of 26,528 Watts at generation 12 (note that there is a near-optimal power point of 26,526 Watts which was found at generation 11). With *idum* set to -20000, the GA found the optimal power of 26,528 Watts at generation 5. Clearly the random number seed can have a big difference on how quickly the GA finds the optimal power for this problem. However, the average power of the population follows similar paths for each random number seed and both seeds result in an average normalized power oscillation around 0.97.

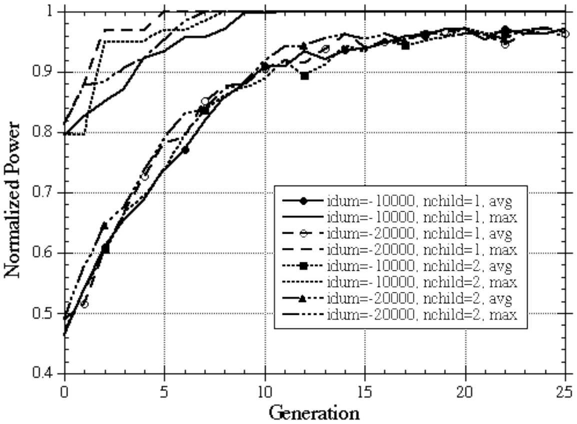

Fig. 3 GA performance versus initial random number seed and number of childeren per pair of parents. For all cases a tournament-uniform-creep-niching-elitist GA with a population size of 100 was used.

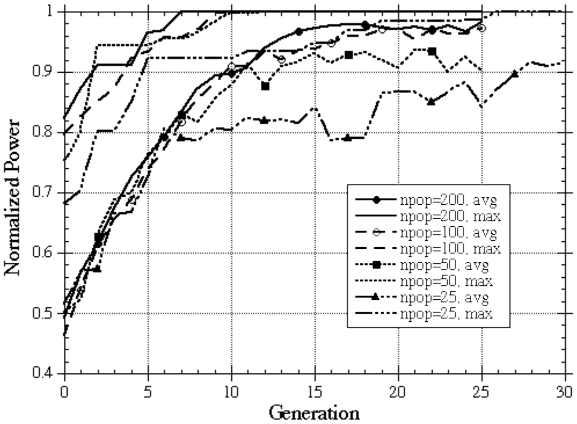

Fig. 4 GA performance versus population size. For all cases a tournament-uniform-creep-niching-elitist GA with idum=-10000 was used.

One or two children, honey? The question of family size is a relatively new one to the GA world. Typically researchers have used two children per pair of parents. During the initial coding of the GA for these studies only one child per pair of parents (*nchild*=1) was used, but it made for an interesting test to see how a larger average family size would evolve, Fig. 3. The initial GA run with *nchild*=2 and *idum*=-10000 found the optimal power point at generation 9 (the near-optimal point at generation 8). Since the *nchild*=1 GA with *idum*=-10000 found the optimal point at a later generation (generation 12), this suggested that perhaps a larger average family size was better. However, a second GA run with *nchild*=2 and *idum*=-20000 found the optimal power point at generation 8 (the near-optimal point at generation 7), i.e., with *idum*=-20000, the *nchild*=1 case provided better performance than the *nchild*=2 case. Thus, at first glance it appears that the GA can perform better or worse with one or two children depending upon the initial random number seed, i.e. the GA performance may be more influenced by the random number seed than by the number of children per pair of parents.

Now let us take a second glance at these results. With *nchild*=2, the optimal power was found in generations 9 and 8 for random number seeds of -10000 and -20000, respectively. With *nchild*=1, the optimal power was found in generations 12 and 5 for random number seeds of -10000 and -20000, respectively. The results of using different random number seeds and *nchild*=2 were more similar than those having *nchild*=1, i.e., there was more consistent GA performance with *nchild*=2 than with *nchild*=1. Therefore, it may be the case that the larger family size (*nchild*=2) adds more stability and less variability to the evolution of the population.

Is bigger better? In general, larger populations should find the optimal individual for the environment in fewer generations than smaller populations. But, at the same time, larger populations take a longer amount of time to compute their progress. For quick function evaluations which take considerably less than a CPU second, it may be acceptable to run large populations. However, for function evaluations which are represented by running a code the size of Blaze II (which takes 30-60 CPU sec for a single calculation), total run time for many generations can be between a day and more than a week depending on the population size.

As expected, Fig. 4 shows that the larger the population size (n_{pop}), that a fewer number of generations are required to find the optimal power; for a population size of 200, the near-optimal power was found at generation 7 and the optimal at generation 8; for a population size of 100, the near-optimal power was found at generation 11 and the optimal at generation 12; for a population size of 50, the near-optimal and optimal powers were both found at generation 12. A more important question is, how many function evaluations (calls to Blaze II) were made to find the optimal power for the different populations? To find the optimal power there were 1600 function evaluations made for n_{pop}=200, 1200 function evaluations for n_{pop}=100, and only 600 function evaluations for n_{pop}=50. Thus, it is clear that population sizes of more than 50 are not necessary to optimize the power for this chemical laser problem. In fact, the n_{pop}=50 case accomplished the optimization task in significantly fewer function evaluations than the larger population sizes. A population size of 25 was also tested and the optimal power was found at generation 26; this corresponds to 650 function evaluations, therefore the n_{pop}=50 case gave slightly better performance.

Of note is that in the first application of GAs to chemical laser problems,(Carroll, 1996) a population size of 50 tended to remove alleles too rapidly from the population base; for that application a population size of 50 was judged too small. However, in those numerical tests both floating-point coding and single-point crossover were being used for the n_{pop}=50 trials. It was also found by Carroll (1996) that uniform crossover tended to preserve alleles better than single-point crossover. Further, Goldberg (1991b) showed that floating-point coded GAs can be "blocked" from finding the global minimum because important alleles can be lost as separated local minimum are found. Thus, it is possible that the use of floating-point coding and single-point crossover was the cause of too rapid an allele loss in the previous chemical laser GA modeling n_{pop}=50 trials. A population size of 50 is quite acceptable with the use of binary coding and uniform crossover in this chemical laser optimization study.

To niche or not to niche, that is the question. The first use of the GA for chemical laser problems involved a highly multimodal parameter space (Carroll, 1995a; Carroll, 1996) and the a priori assumption was that this chemical laser optimization problem would also be highly multimodal. Since niching (or sharing) has been shown to be an effective GA technique for multimodal problems, Goldberg's (1987) multidimensional phenotypic sharing scheme with a triangular sharing function was implemented. (Note that to find the multidimensional distance from the best individual in the population, all parameter differences were normalized.) The loaded-GA of this study included niching and the baseline n_{pop}=100, *nchild*=1, *idum*=-10000 run found the optimal power at generation 12. When niching was turned off in the loaded-GA, the near-optimal power point (26,526 Watts) was found sooner at generation 7, but the actual optimal power point (26,528 Watts) was not found until generation 13, Fig. 5. As it turns out for this problem, there is not the highly multimodal structure to the parameter space as there was for the previous chemical laser GA application,(Carroll, 1995a; Carroll, 1996) thus niching may actually hinder the climbing of a single major peak. This may be the reason that the near-optimal power point was found more rapidly when niching was removed from the GA. The fact that the non-sharing GA took another 6 generations to go from near-optimal to optimal is most likely a random effect. Since chemical laser problems will often be multimodal, it is felt that niching (sharing) is a useful technique for this application of GAs.

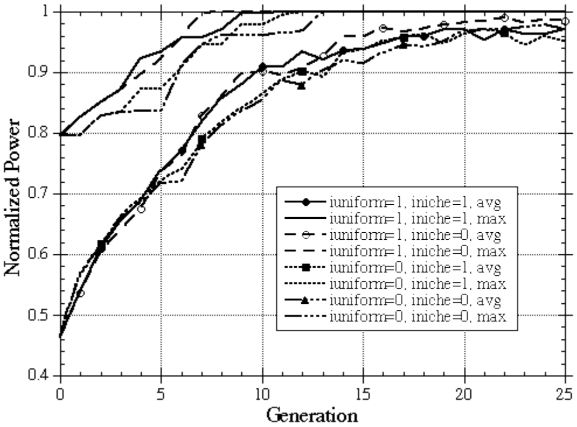

Fig. 5 GA performance versus niching and crossover scheme. For all cases a tournament-creep-elitist GA with a population size of 100 and idum=-10000 was used.

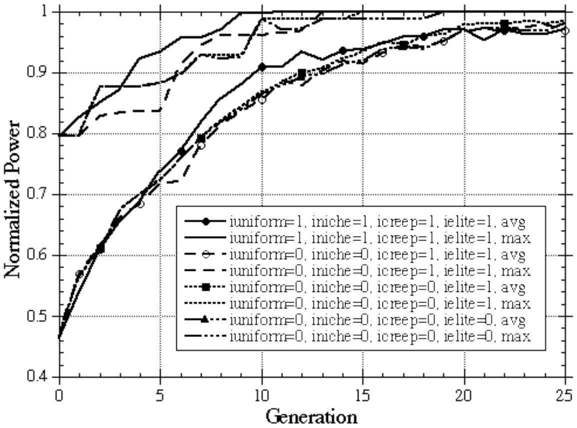

Fig. 6 GA performance versus creep mutations and elitism. For all cases a tournament GA with a population size of 100 and idum=-10000 was used.

Crossroads. As mentioned above, Carroll (1996) found that uniform crossover (Syswerda, 1989) tended to preserve more alleles than single-point crossover. For that reason, uniform crossover was the preferred choice in that work and also the motivation for using uniform crossover in the loaded-GA of this work. However, the question is raised as to whether or not there is a significant difference between the two crossover choices for this application? Figure 5 shows that the uniform crossover case (*iuniform*=1, *iniche*=1) approaches the optimal solution more rapidly than the single-point crossover case (*iuniform*=0, *iniche*=1), but actually they both obtain the optimal power at generation 12. As for the allele pool, both cases end up oscillating around an average normalized power of approximately 0.96, therefore neither crossover scheme appears to show more or less allele preservation for this problem. Since the uniform crossover case initially approaches the optimal solution more rapidly, there is still a slight personal preference for uniform over single-point crossover.

Uniform sharing. One of the important attributes of uniform crossover and sharing (niching) is that both of these GA techniques tend to preserve variety in the genetic soup. What happens to the loaded-GA performance when both of these allele preservers are removed? Figure 5 shows that when both uniform crossover and niching are turned off, the GA performance slows down. In fact, without uniform sharing (*iuniform*=0, *iniche*=0), the near-optimal power was not found until generation 13 and the optimal power was not obtained until generation 17 (whereas with uniform sharing the near-optimal and optimal powers were found at generations 11 and 12, respectively). Thus, it appears that uniform crossover and niching supplement each other slightly and it is important to have at least one of these techniques in the GA for better performance.

Creepy crawly. Creep mutations can be useful in the sense that they can slide your gene pool toward the optimal solution rather than just having to jump towards it. Figure 6 compares the non-uniform, non-niching, creeping case (*iuniform*=0, *iniche*=0, *icreep*=1, *ielite*=1) with the non-uniform, non-niching, non-creeping case (*iuniform*=0, *iniche*=0, *icreep*=0, *ielite*=1). With creep mutations removed, the GA did not find the near-optimal power until generation 15, but still managed to find the optimal power at generation 17 (the same generation as the creeping case). Since the addition of creep mutations found the near-optimal power sooner than without them, creep mutations appear to be of slight benefit to the GA.

King of the hill. Elitism forces the best individual (the king) to remain at the top of the hill until someone (from the king's own proginy or an usurper from the throng) comes along who is strong enough to dethrone the king. When elitism is removed, the king must come down from the hill every generation, breed with the throng and hope his proginy can reclaim the throne. Elitism was important for the previous chemical laser GA application;(Carroll, 1996) is it also beneficial for this application? The basic tournament selection GA with elitism (*iuniform*=0, *iniche*=0, *icreep*=0, *ielite*=1) is compared with the basic tournament selection GA without elitism (*iuniform*=0, *iniche*=0, *icreep*=0, *ielite*=0) in Fig. 6. When elitism was removed the GA performance was inhibited. The near-optimal power was found at generation 19 and the optimal power at generation 20. Clearly elitism is beneficial for this application.

How about really, really small? Krishnakumar (1989) found that a micro-GA (or µGA) avoided premature convergence and demonstrated faster convergence to the near-optimal region than did a simple GA (SGA) for the multimodal problems he studied. Krishnakumar’s results are quite intriguing for the multimodal, CPU intensive chemical laser application of this study. Very briefly, a µGA starts with a random, very small population, the population evolves in normal GA fashion and converges in a few generations (typically 4 or 5); at this point, a new random population is chosen while keeping the best individual from the previously converged generation and the evolution process restarts. For this study, population convergence is defined to occur when less than 5% of the bits of the other individuals are different than the best individual, i.e., convergence occurs when on average 1 bit (or fewer) of the 22 per individual is different from those of the king of the hill.

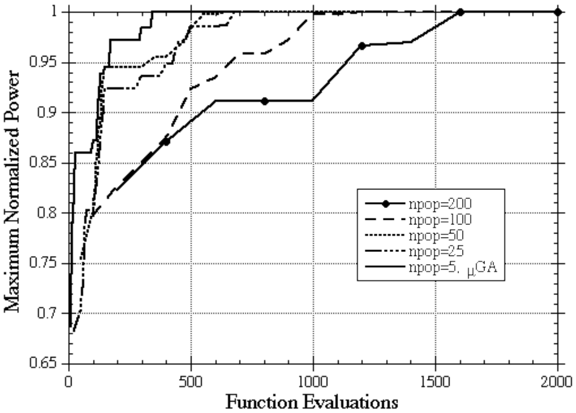

Fig. 7 Number of function evaluations as a function of population size and GA scheme. A normalized fitness of 1.0 corresponds to the optimal power of 26,528 Watts.

Krishnakumar’s µGA used n_{pop}=5, tournament selection, no mutations, elitism and single-point crossover with p_{cross}=1.0; using these µGA parameters, the optimal power of 26,528 Watts was found at generation 65 (= 325 function evaluations). Switching to uniform crossover gave roughly the same performance with the µGA; the optimal power was found at generation 69

(= 345 function evaluations). However the uniform-μGA is able to handle an order-3 deceptive function which the single-point crossover μGA was not able to optimize, Appendix A; this indicates that the uniform-μGA is more robust than the single-point crossover μGA. Figure 7 shows a comparison of the uniform-μGA results with those of the loaded-GA having different population sizes; it is clear that the uniform-μGA reaches the optimal power faster than the loaded-GA for any population size. Note that average population fitness values are not meaningful with a μGA because of the start-restart nature of the μGA evolution process.

Different micro-population sizes were also tested. The uniform-μGA with n_{pop}=3 found the optimal power at generation 268 (= 804 function evaluations) and the uniform-μGA with n_{pop}=10 found the optimal power at generation 65 (= 650 function evaluations). Thus, it appears that Krishnakumar’s choice of 5 for a population size is roughly optimal for this application.

Since the uniform-μGA obtains the optimal power in a significantly fewer number of function evaluations than the best loaded-GA case (n_{pop}=50), 345 evaluations versus 600, it is clear that the uniform-μGA method gives better performance. The uniform-μGA has two additional benefits, (i) there is no need to fiddle with GA operators such as mutation probabilities, and (ii) it is able to handle a loosely ordered order-3 deceptive function which the more traditional GA methods (including the loaded-GA) and the single-point crossover μGA were not able to optimize, Appendix A. While this particular chemical laser GA application is not very deceptive, the uniform-μGA is highly recommended for future chemical laser GA applications which are potentially deceptive in nature.

V. Concluding Remarks

With the Blaze II model baselined to RADICL gain data, a genetic algorithm (GA) was used to predict optimal flow conditions and device configuration at high total pressures in the range of 100-250 Torr. This is the first known application of the genetic algorithm technique for optimizing the performance of a laser system (chemical, solid-state, or gaseous). An optimal set of flow conditions and geometry was found with a predicted power of 26.5 kW; since the baselined predicted powers were approximately 33% higher than were measured powers,(Carroll, 1995b) the likely measured power should be roughly 19.9 kW. The total pressure of this flow is approximately 100 Torr. The optimal flow conditions maximized the Cl_2 (or equivalently the O_2) flow rate. The power was maximized by minimizing the primary He flow rate to a level which still maintained a high stagnation pressure of 100 Torr; lower He flow reduces He deactivation and lower pressure improves diffusional mixing. For high Cl_2 flow rate conditions, minimizing the titration ratio to values of $\beta \approx 0.010$ maximized the power. Maximizing the secondary He flow improved the power by increasing the penetration and mixing. The predicted Fabry-Perot power was maximized by moving the resonator leading edge downstream. Power was optimized with larger throat sizes and with no additional spacer plate between injectors and throat; this suggests that it may be possible to improve power with larger throats and/or decreasing the distance between the injectors and the throat. An increase from 3° to 4.5° in the constant expansion angle cavity section is recommended, but is predicted to improve the power by only 1-2%. It is important to point out that conclusions based on these trends may only be appropriate for the studied flow regime. Most importantly, this modeling indicates that it should be possible to obtain significant power levels (> 19 kW) with the RADICL device under high pressure operating conditions.

Several different genetic algorithm techniques were tested for this optimization problem. The initial random number seed can have a big difference on how quickly the GA finds the optimal power for this problem. The GA can perform better or worse with one (*nchild*=1) or two (*nchild*=2) children per pair of parents depending upon the initial random number seed, however it may be the case that the larger family size (*nchild*=2) adds more stability and less variability to the evolution of the population. Population sizes of more than 50 are not necessary to optimize the power for this chemical laser problem. In fact, the n_{pop}=50 case accomplished the optimization task in significantly fewer function evaluations than the larger population sizes.

Since chemical laser problems will often be multimodal, it is felt that niching (sharing) is a useful technique for this application of GAs (despite the fact that it made an insignificant difference for this particular application). While the differences between uniform and single-point crossover were not very significant in this study, since the uniform crossover case initially approaches the optimal solution more rapidly, there is a slight personal preference for uniform over single-point crossover. More significantly, it appears that uniform crossover and niching supplement each other slightly and it is important to have at least one of these techniques in the GA for best performance. Since the addition of creep mutations found the near-optimal power sooner than without them, creep mutations appear to be of slight benefit to the GA. Elitism is beneficial for this application. The best quasi-traditional GA for this application uses binary coding, tournament selection, uniform crossover, jump and creep mutations, niching (sharing) and elitism. A population size of 50 is suggested for this problem because this size found the optimal power in the fewest number of function evaluations for the loaded-GA. There seems to be little difference in GA performance when using one or two children per pair of parents, although the two children case did have less variability in the evolution of the population when using different initial random number seeds.

The recommended technique for future chemical laser applications is the μGA using uniform crossover with a population size of 5. The uniform-μGA reached the optimal power in only 345 function evaluations which is faster than the loaded-GA for any population size. Different micro-population sizes were tested and the results indicate that a micro-population size of 5 provides roughly the optimal performance. The uniform-μGA has two additional benefits, (i) there is no need to fiddle with GA operators such as mutation probabilities, and (ii) it is able to handle a loosely ordered order-3 deceptive function which the more traditional GA methods (including the loaded-GA) and the single-point crossover μGA were not able to optimize. For Goldberg's (1989b) order-3 deceptive function, the uniform-μGA gave comparable average performance to that of the robust and sophisticated messy-GA (mGA); to the author's knowledge, the uniform-μGA has not previously been tested on this deceptive function. It must be noted that while the uniform-μGA has demonstrated good performance for this order-3 deceptive function, it may or may not provide good performance for much harder order-5 deceptive functions. Regardless, these results indicate that the somewhat simpler uniform crossover μGA is a reasonably robust and efficient GA technique capable of tackling at least some order-3 deceptive problems.

Acknowledgements

This work was supported in part by the Air Force through Logicon Research and Development Associates subcontract 907629. Many thanks to D. Goldberg for several useful genetic algorithm conversations.

References

Barmashenko, B.D., Elior, A., Lebiush, E., and Rosenwaks, S., 1994, "Gain and Power in COILs - Theory and Experiment," AIAA Paper 94-2434.

Buggeln, R.C., Shamroth, S., Lampson, A.I., and Crowell, P.G., 1994, "Three-Dimensional (3-D) Navier-Stokes Analysis of the Mixing and Power Extraction in a Supersonic Chemical Oxygen Iodine Laser (COIL) with Transverse I_2 Injection," AIAA Paper 94-2435.

Carroll, D.L., 1995a, "Modeling High-Pressure Chemical Oxygen-Iodine Lasers," *AIAA J.*, Vol. 33, 8, pp. 1454-1462.

Carroll, D.L., 1995b, "Optimizing High Pressure Chemical Oxygen Iodine Lasers," International Conference on Lasers '95, Charleston, South Carolina, Dec. 4-8.

Carroll, D.L., 1996, "Chemical Laser Modeling with Genetic Algorithms," *AIAA J.*, Vol. 34, 4, in press.

Crowell, P.G., and Plummer, D.N., 1993, "Simplified chemical oxygen iodine laser (COIL) system model," *Intense Laser Beams and Applications*, International Society for Optical Engineering, Vol. 1871, Los Angeles, CA, pp. 148-180.

Dymek, A., 1992, "An examination of hypercube implementations of genetic algorithms," M.S. and Report No. AFIT/GCS/ENG/92M-02 Thesis, Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio.

Eshelman, L.J., 1989, "The CHC Adaptive Search Algorithm: How to Have Safe Search When Engaging in Nontraditional Genetic Recombination," in: *Foundations of Genetic Algorithms*, Spatz, B. M. (Ed.), Morgan Kaufmann, San Mateo, CA, pp. 265-283, 1991.

Goldberg, D.E., and Richardson, J., 1987, "Genetic Algorithms with Sharing for Multimodal Function Optimization," Genetic Algorithms and their Applications: Proceedings of the Second International Conference on Genetic Algorithms, pp. 41-49.

Goldberg, D.E., 1989a, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading, MA.

Goldberg, D.E., Korb, B., and Deb, K., 1989b, "Messy Genetic Algorithms: Motivation, Analysis, and First Research," in: *Complex Systems 3*, Complex Systems Publications, pp. 493-530.

Goldberg, D.E., and Deb, K., 1991a, "A Comparitive Analysis of Selection Schemes Used in Genetic Algorithms," in: *Foundations of Genetic Algorithms*, ed. by Rawlins, G.J.E., Morgan Kaufmann Publishers, San Mateo, CA, pp. 69-93.

Goldberg, D.E., 1991b, "Real-coded Genetic Algorithms, Virtual Alphabets, and Blocking," in: *Complex Systems*, Vol. 5, Complex Systems Publications, pp. 139-167.

Goldberg, D.E., Deb, K., and Clark, J.H., 1992, "Genetic Algorithms, Noise, and the Sizing of Populations," in: *Complex Systems*, Vol. 6, Complex Systems Publications, pp. 333-362.

Goldberg, D.E., Deb, K., Kargupta, H., and Harik, G., 1993, "Rapid, Accurate Optimization of Difficult Problems Using Fast Messy Genetic Algorithms," in: *Proceedings of the Fifth International Conference on Genetic Algorithms*, Forrest, S. (Ed.), Morgan Kaufmann, San Mateo, CA, pp. 56-64.

Helms, C.A., Shaw, J., Hager, G.D., Truesdell, K.A., Plummer, D.N. and Copland, R.J., 1994, "Iodine Dissociation in COILs," AIAA Paper 94-2437.

Helms, C. A., 1995, private communication, Phillips Laboratory, Kirtland AFB.

Holland, J. H., 1975, 1992, *Adaptation in Natural and Artificial Systems*, A Bradford Book, The MIT Press, Cambridge, MA.

Krishnakumar, K., 1989, "Micro-Genetic Algorithms for Stationary and Non-Stationary Function Optimization," SPIE: Intelligent Control and Adaptive Systems, Vol. 1196, Philadelphia, PA.

Madden, T.J., Carroll, D.L., and Solomon, W.C., 1994, "Detailed Mixing in COIL Devices," AIAA Paper 94-2432.

Muhlenbein, H., 1992, "How genetic algorithms really work: I. Mutation and Hillclimbing," in: *Parallel Problem Solving in Nature, 2*, Manner, R., Manderick, B. (Eds.), North-Holland, Amsterdam, pp. 15-25.

Scott, J.E., Shaw, J.L.R., Truesdell, K.A., Hager, G.D., and Helms, C.A., 1994, "Design Considerations for the Chemical Oxygen-Iodine Supersonic Mixing Nozzle," AIAA Paper 94-2436.

Sentman, L.H., Subbiah, M., and Zelazny, S.W., 1977, "Blaze II: A Chemical Laser Simulation Computer Program," Bell Aerospace Textron, Buffalo, NY, T.R. H-CR-77-8.

Syswerda, G., 1989, "Uniform Crossover in Genetic Algorithms," in: *Proceedings of the Third International Conference on Genetic Algorithms*, ed. Schaffer, J., Morgan Kaufmann, Los Altos, CA, pp. 2-9.

Tate, R.F., Hunt, B.S., Helms, C.A., Hager, G.D., and Truesdell, K.A., 1994, "2-D Gain Measurements in a COIL Device," AIAA Paper 94-2438.

Truesdell, K.A., Lamberson, S.E., and Hager, G.D., 1992, "Phillips Laboratory COIL Technology Overview," AIAA Paper 92-3003.

Whitley, L.D., 1991, "Fundamental principles of deception in genetic search," in: *Foundations of Genetic Algorithms*, Spatz, B. M. (Ed.), Morgan Kaufmann, San Mateo, CA, pp. 221-241.

Appendix A: μGAs and an Order-3 Deceptive Function

While the loaded-GA and μGAs gave excellent performance for this application, a natural question is, how robust are these GAs for other applications? To truly test their ability to handle difficult problems, Goldberg's (1989b) order-3 (3 bits per parameter) deceptive function, was tested. A deceptive function is defined as "one in which low-order schema fitness averages favor a particular local optimum, but the global optimum is located at that optimum's complement."(Goldberg, 1989b) For this particular function (which has been used by several researchers (Dymek, 1992; Eshelman, 1991; Muhlenbein, 1992; Whitley, 1991), there are 10 parameters, each with 3 bits having a value given by

bit	value	bit	value
111	30	100	14
101	0	010	22
110	0	001	26
011	0	000	28

The global optimum is a string of thirty 1's (fitness=300), while the favored next highest local optimum is a string of thirty 0's (fitness=280). This function has 1024 local optima from a Hamming space viewpoint, or 59,049 optima in a ten-dimensional real space.(Goldberg, 1989b) The problem is further complicated by the use of tight or loose bit ordering. In tight ordering, the 3 bits of each parameter are adjacent to one another such that the defining length of each building block is short ($\delta = 2$), e.g., the bits of the first parameter are located in positions 1, 2 and 3 in the chromosome string. With loose ordering, the 3 bits of each parameter are separated from one another such that the defining length of each building block is long ($\delta = 20$), e.g., the bits of the first parameter are located in positions 1, 11 and 21 in the chromosome string. With single-point crossover it is only possible to break up one building block in the tight ordering scheme, but it is possible (and likely) to break apart many building blocks with loose ordering; this should (and does) make it extremely difficult for single-point crossover to find the global optimum in the loosely ordered problem.

To summarize Goldberg's (1989b) results, a simple GA with tournament selection located the global optimum after 40,000 function evaluations with tight ordering, but was completely deceived with loose ordering and converged to the local optimum with fitness of 280. The robust and sophisticated messy-GA (mGA) found the global maximum for both orderings after 40,600 function evaluations.(Goldberg, 1989b) Before testing the loaded-GA and μGA schemes, the simple GA with tournament selection results were duplicated with the GA of this paper using Goldberg's (1989b) parameters (n_{pop}=2000, p_{cross}=1.0, no mutations).

When the loaded-GA with single-point crossover was tested, similar results to those of the simple GA were obtained, i.e., the global optimum was located at roughly 40,000 function evaluations with tight-ordering, but was deceived to one of the many local optimum with loose ordering. The loaded-GA with uniform crossover was then tested for both tight and loose ordering and interestingly was deceived to local optima with both orderings. The reason that the loaded-GA with uniform crossover failed is believed to be a consequence of the fact that uniform crossover will indiscrimanately destroy building blocks and allow the GA to climb the favored hill to the deceptive optimimum; with single-point crossover, the tight ordered 111 building blocks could remain intact. For the traditional GA approach, it appears that uniform crossover is not of benefit in deceptive problems.

The single-point crossover μGA gave very impressive results for the tightly ordered case; the global optimum was found after an average of 9,750 function evaluations for three different random number seeds. However, when loose ordering was used, the single-point crossover μGA was always deceived to one of the many local optima. Elitism, which is invoked in the μGA scheme, was sufficient to retain some of the early developed 1...1...1 building blocks with loose ordering, but once a fairly fit individual was established, single-point crossover would destroy the subsequent development of other 1...1...1 building blocks.

The uniform crossover μGA also produced impressive convergence for the tightly ordered case; the global maximum was found after an average of 21,633 function evaluations for three different random number seeds. While this was somewhat worse than for the single-point crossover μGA, it was better performance than that of the simple GA. More importantly, when the loosely ordered case was tested, the uniform- μGA found the global optimum after an average of 36,805 function evaluations for three different random number seeds; to the author's knowledge, the uniform-μGA has not previously been tested on this deceptive function. Thus, the uniform- μGA gave comparable average performance to that of the robust and sophisticated mGA as well as Eshelman's (1991) CHC non-traditional GA. The uniform-μGA is in some ways similar in character to CHC because both schemes invoke elitism, uniform crossover, and population restart concepts; however, the uniform-μGA uses a smaller population (5 versus 50), many more population restarts, and is somewhat simpler to implement. It must be noted that while the uniform-μGA has demonstrated good performance for this order-3 deceptive function, it may or may not provide good performance for much harder deceptive functions such as Goldberg's (1993) order-5 deceptive function. Regardless, these results demonstrate that the somewhat simpler uniform crossover μGA is a reasonably robust and efficient GA technique capable of tackling at least some order-3 deceptive problems.

Why was the uniform-μGA able to tackle this deceptive problem? The answer to this is partly due to the constant infusion of new genetic information as the micro-population is reborn (this is where the simple and loaded GAs fail), and partly because uniform crossover does not care where the components of building blocks are located (this is where single-point crossover fails); Syswerda (1989) and Eshelman (1991) have previously pointed out that the uniform crossover operator is not position biased. In other words, good new genetic information (1 bits) can be inserted into strong members of the population without having to break apart other good building blocks. Further, with elitism invoked, the worst that can happen in a generation is to keep the same best member; this prevents the loss of the good building blocks and allows the μGA to climb its way to the top of the steepest hill without stepping backwards towards the favored hill. One might expect that the loaded-GA with elitism and uniform crossover might have a chance at finding the global maximum because it is possible to argue that mutation represents a constant infusion of new genetic information; however, at the point where there are as many 1...1...1 building blocks as 0...0...0 building blocks, mutation will tend to destroy as many building blocks as it creates and this will stick the loaded-GA at some local optimum, but not the global optimum.