COSMO: A Correlation Sensitive Mutation Operator for Multi-Objective Optimization

Varun Aggarwal
Computer Science and Artificial Intelligence Lab
Massachusetts Institute of Technology
Cambridge, MA
varun_ag@mit.edu

Una-May O'Reilly
Computer Science and Artificial Intelligence Lab
Massachusetts Institute of Technology
Cambridge, MA
unamay@csail.mit.edu

ABSTRACT

This contribution is the first to discover exploitable structural features within circuit optimization problems (COP) and discuss how it is indicative of a general structure and possibly a 'measure of hardness' in real-world multi-objective optimization problems. We then present a methodology to exploit this structure in a multi-objective evolutionary algorithm by designing a novel Correlation Sensitive Mutation Operator, COSMO. COSMO is, at the least, universally applicable in the domain of circuits and we discuss how it can be easily extended to other domains. We discuss the rationale behind COSMO and interpret it in context of dimensional locality. We compare COSMO's performance with the traditional operators used for multi-objective optimization. For two instances of circuits, we show that COSMO gives significantly faster and better optimization than conventional operators. The paper also takes the first steps in thinking and interpreting how operators for MO-EAs should be designed.

Categories and Subject Descriptors

G.1.6 [**Numerical Analysis**]: Optimization-Global Optimization, Nonlinear programming; I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search

General Terms

Algorithms, Human factor, theory

Keywords

Evolutionary, multi-objective optimization, operators, fitness landscape, circuits, monotonic functions

1. INTRODUCTION

GAs are deployed successfully with only deft problem transfer and conventional operators and representation because the problem has structure that well matches them.

GECCO'07, July 7–11, 2007, London, England, United Kingdom.
Copyright 2007 ACM 978-1-59593-697-4/07/0007 ...$5.00.

Members of the genetic algorithm (GA) community perpetually face the need to understand structure in the specific domain of their problem and design a GA that exploits it. By understanding the nature of the structure, an appropriate genotype representation and variation operator [1] can be designed and the adaptive search process can be set up to converge to better solutions more efficiently. At present, for multi-objective optimization Evolutionary Algorithms (MO-EAs) with real-valued variables, a well rationalized theory of how to design variation operators is lacking.

There has been considerable interest in multi-objective (MO) optimization of analog circuits (discussed in Section 3) in recent times. As demonstrated by the marginal adequacy of current state of art MO-EAs,circuit sizing is not a needle in a haystack problem. We asked the question, whether there is any structure in the circuit search space which can be exploited by MO-EA operators for efficient operators. This question also begs the need of a theory for how operators should be designed for MO-EAs.

This contribution is the first to discover exploitable structural features within a circuit optimization problem (COP) and translate this to effective, novel MO-EA operator, which we call COSMO (Correlation Sensitive Mutation Operator). We show through experiments that optimization is much more accurate as a result of using these operators without incurring any additional computational cost (fitness evaluations). We compare with the traditionally used SBX (simulated binary crossover) and polynomial mutation [7].

Apart from designing efficient operators for multi-objective optimization of circuits, we make the following general contributions. We make a claim on the kind of structure multi-objective optimization may in general possess. This can be instructive towards design of efficient MO-EAs. We develop a hypothesis for how COSMO works which leads to a general suggestion on a methodology to design operators for MO-EAs. We believe that these are the first steps in thinking about operators for MO-EAs and the methodology will transfer to MO problems in other domains. We provide our rational for predicting this in the paper.

In Section 2, we consider the different rationales behind the current array of variation operators employed in genetic algorithms, evolutionary strategies, particle swarm optimization and Estimation of Distribution algorithms (EDAs). Extensions of these rationales for MO-EAs are lacking among the various algorithms. In Section 3, we discuss the COP

[1] Representation and variation operator have no independent meaning. Their meaning is interdependent. For future discussion, we use the term variation operators to imply both.

and identify exploitable structure in it. We argue why this structure may generalize to MO problems in other domains. In Section 4, we describe COSMO which exploits the identified structure for efficient optimization. We give an intuitive rationale for COSMO followed by a perspective on it from the view-point of variation operators in EAs. In Section 5, we present results of implementing COSMO into NSGA-II which show that, without incurring any additional computational cost, circuit optimization is more accurate as a result of using COSMO. Finally, we conclude the paper in Section 6.

2. VARIATION OPERATORS

In this section, we present a perspective of how variation operators are understood and interpreted in the EA community with respect to addressing problems with real valued or discrete variables.

Discrete problems: Genetic algorithms with discrete encodings have enjoyed considerable success in solving discrete problems. Two variation operators: crossover and mutation, are used. The understood role of a variation operator is based on the building block hypothesis. The crossover operator's purpose is to combine the *good parts* (building blocks) of the parent chromosomes to form fitter offspring. The mutation operator plays a role that investigates small variations around possible building blocks. The initially popular crossover operators were one-point and two-point crossovers. The crux of the variation issue is how the algorithm can identify and promote the building blocks in the chromosome. In case of one-point or two-point crossover, building block are not preserved when correlated variables are not spatially aligned or may have multiple interactions. Estimation of variable correlation statically and tailoring variation operators to exploit them has been done to solve artificially constructed problems to validate the building-block hypothesis [11]. Since this tactic assumes often unavailable information, approaches that dynamically learn some *structure* (or equivalently building blocks) from the selected fitter solutions have been developed. Here offspring are created to partially preserve an observed (i.e. learned) structure that is characteristic of the fitter solutions. Estimation of Distribution Algorithms (EDA) or Probabilistic Model Building Genetic Algorithms (PMGA) [19] are such examples. In fact EDAs do not employ conventional variation operators. In lieu of directly varying a parent (or two parents) to generate a variant, they learn a model of structure and sample this model to create offspring that vary along the identified structure. In some cases, this structure could be expressed in terms of learning crossover boundaries connecting back to the GA theory. EDAs can solve many difficult problems that an EA with standard variation operators cannot, but they incur a heavy expense for the effort to accurately construct a model of variable correlation. Nonetheless they illustrate how critical the variation step (or knowledge of structure) is for the success of an EA.

Real-valued problems: For real valued optimization, genetic algorithms, evolutionary strategies (ES) and evolutionary programming (EP) each offer an approach to variation[2]. Many GAs encode a real-valued variable with a discrete encoding such as binary, BCD, or gray coding. Variation operators are blindly applied on the discrete encoding. The blind nature of the variation is intentional. The motivation arises from evidence in nature that genetic variation is often random. Later, Deb devised a simulated binary crossover, SBX [9], which numerically simulates the effect of bitwise crossover on a pair of binary encoded real values, but which does not require encoding a real number in binary.

One might argue that, while these kinds of GA honor natural genetic variation, they forfeit an opportunity to exploit any structure the given problem may have. This counter-rationale and the traditional optimization method perspective that considers *search trajectory* are observable in ES and EP [2]. In these approaches, covariance or standard deviation of the mutation operator are co-evolved during the run. Also, crossover can create the average of the parents' variables. These same rationales extend to particle swarm optimization (PSO) [16] and Covariance Matrix Adaptation Evolutionary Strategies (CMA-ES) [12]. PSO simulates local trajectory guidance mixed with global influence, while CMA-ES tries to align the search trajectory to an approximated Newton direction (by learning it from selected individuals). CMA-ES works very well on a number of hard optimization problems. When EDAs address real-valued optimization, they attempt to learn the structure of real-valued promising solutions with a probability distributions but reliably, good models are, to date, generally elusive (see [3]). The real-valued EDAs that have demonstrated good performance are similar to CMA-ES, and indicate a convergence in approaches.

Real-valued Multi-objective Optimization: For real-valued MO optimization algorithms, primarily SBX crossover and polynomial mutation have been used (in NSGA-II [8] and SPEA2 [21]). Pareto-archived Evolutionary Strategies (PAES) [17] use random mutation and do not co-evolve mutation variances as an ES does. The rationale for these blind variation methods might be purposeful. On the other hand, it might be the result of simple, direct transfer.

An extension of a multi-objective optimization to incorporate any existing variation strategy is straightforward to design and evaluate. The variation step of a MO-EA can be mechanically replaced with the variation step of any desired algorithm, while retaining the Pareto selection step. Real-valued EDAs [19] and CMA-ES [14] have been extended for multi-objective optimization in similar ways. However, it must be considered that the re-used variation operators are designed for optimizing a single objective (leading to a single optima in most cases), while a MO problem has multiple objectives and a set of pareto-optimal points as the solution. A MO-EA shall need multiple trajectories as opposed to a single one as in case of single-objective optimization (also noted in [14]). This makes such mechanical transfers short of a strong rationale or theory. For instance, CMA-ES variation trajectory directs the search in the Newton's direction towards of a single point and not a multiplicity of Pareto-optimal solutions. Some works [15] show that in presence of "connectedness and regularity", the search can be done in two phase, first directed towards a single point and in the second phase, an exploration in the vicinity of the point.

It becomes apparent that despite the importance and value of identifying and exploiting structure, real-valued MO-EAs have, to date, not focused on it. Current variation operators are mostly rote transfers from single objective approaches. It remains open to question if we can understand more about variation in MO-EA from the point of view of multiple optimal solutions, multiple objectives and their interaction.

We now turn to circuit sizing where current EA and MO-

EA approaches still fall tantalizingly short of offering a clearly superior solution. In the domain of circuits, might knowledge of structure be helpful to an MO-EA? Because the nature of the structure in circuits seems to be quite general, it begs the question of whether there are generalizable heuristics for real-valued MO-EA variation operators for application domains with similar, such general structure?

3. STRUCTURE OF COP

3.1 Circuit Optimization Problem

An analog circuit has 10 to 200 real-valued parameters that must be set in order to meet its specifications (between 2 to 20 in number). The process of setting these parameters is called *circuit sizing*. Automatic analog circuit sizing is a problem of major importance to the Electronic Design Automation and System-On-Chip design industry [13].

Historically, circuit sizing has been done manually by human designers. More recently, a variety of of approaches for automatic analog circuit sizing have emerged. Circuit sizing using MO-EAs including NSGA-II [5], SPEA-II [10], a hybrid of genetic algorithm and simulated annealing (SA) [20], and a parallel SA [13] are employed both by academics and industry engineers. In the evolutionary algorithm (EA) based approach to sizing, circuit specifications are expressed as objectives and constraints. Each fitness evaluation is a circuit simulation in SPICE and is computationally very expensive. Because of computational expense, the efficiency and scalability of the algorithm are vital. A typical perspective on convergence performance measures how much a circuit is accurate to specifications for a specified budget of computational resources. Circuit designers begrudgingly acknowledge the worth of an EA because for larger circuits each algorithm inevitably fails: computational resources are in a constant arms race with circuit scale. This has led to widespread implementation of these algorithms on parallel clusters [13].

3.2 Identification of structure

To identify structure in COP, we asked the question as to how does a human designer size a circuit. Designers start by decomposition. Consulting the topology they derive a set of parameterized *approximate* non-linear expressions for the circuit's specifications. These equations are not invertible and cannot be solved for the variables given specification values. Instead, the designer must set some of the variables according to common practice to meet some specifications and approximately solve for the remaining variables. They are still far from done because these expressions are first order approximations of the circuit behavior, the solution is approximate and useful more for intuition than being accurate. The closest (and strongly reliable) estimation of the real behavior of circuit is simulation with SPICE which models the circuit more accurately and use numerical methods for solutions. Therefore, the next step is to check the behavior of the circuit with simulation. Generally the circuit doesn't meet all specifications. This leads to choosing a new set of (hopefully improved) parameter values to meet the specifications. Design toward specifications continues with the iterative process of "set, observe,analyze, refine" as the designer progressively intuits the opaque, complex relationship between the component sizes, topology and circuit behavior.

Complex as the problem is, human designers manage to solve it reasonably well. They use an approach used in other complex domains where parameters influence many specifications: Circuit designers typically vary variables to meet one specification, which leads another specification to fall out. They then vary another set of variables to meet the unmet specification and iterate. The question is why and how this search process converges. The success of the process indicated that there must be some "weak" structure in the search space. An examination of the designers' approximate circuit equations (that express the specifications as a function of the parameters) revealed the following structure:

1. Each design specification is influenced by (correlated to) a small number of design variables according to first-order approximations.

2. Each design specification is monotonic, i.e. non-increasing or non-decreasing with regard to the influencing design variables *to first order approximation.*

These observation answered why the designer's iterative process converged. Had all (or majority of) variable been correlated to multiple specifications, just changing one variable at a time would have derailed progress on multiple other specifications. However, since each specification is correlated with a few design variables, the designer can make progress on one specification without disturbing others. Due to the monotonic relationship between variables and specifications, the designer knows which direction to move the variables to improve the specification. Had this not been the case, the designers would need to re-calculate the sensitivity direction of specification with variables at each iteration.

Opamp Example: To elaborate, an example is helpful (full details are in [1]). We consider a simple two-stage opamp shown in Figure 1. The chosen design variables are $L1$, $L3$, $L5$, $L8$, $V1$, $V2$, $V3$, $Iref$, $I1$, $I2$, R_c and C_c. For matching, $L1 = L2$, $L3 = L4$ and $L6 = L7 = L8$. Let us consider one specification of opamp, i.e. gain for illustration. Gain is expressed as follows:

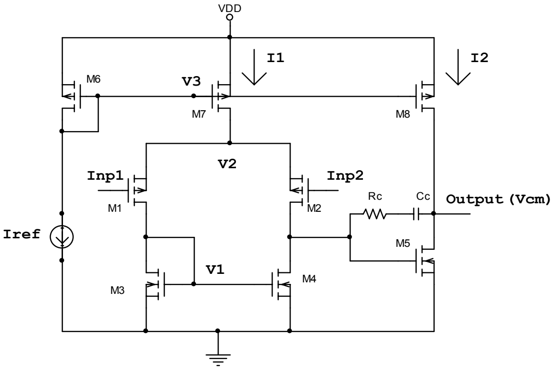


Figure 1: Simple 2-stage Opamp

$$gain = \frac{1}{(Vgs1 - |Vtp|)(Vgs3 - Vtn)(\frac{1}{L1} + \frac{1}{L4} + \frac{1}{L5} + \frac{1}{L8})} \quad (1)$$

Equation 1 clearly shows that gain is non-increasing with $Vsg1$ and $Vgs3$ and non-decreasing with $L1$, $L4$, $L5$ and $L8$.

It is not correlated with the other variables. All other variable correlations and their directional sensitivity with each variable have been derived from similar approximate circuit equations and are presented in Table 1. The columns of the table represent design variables and rows refer to specifications. For each specification, the first row indicates correlation, 1: Design variable is correlated, 0: Design variable is not correlated, 2: Design variable is weakly correlated. The second row indicates whether specification is monotonically increasing (1) or monotonically decreasing (-1) with the design variable. A 1 in the first row and a 0 in the second row indicates that the variable is correlated to the specification, but the specification is not monotonic with it. For this topology each specification is on average correlated to 3.6 variables (out of 12) and all but one specifications are monotonic with all variables. This verifies our hypothesis.

3.3 General structure in MO-EA problems

We believe that the first structural property, i.e. each design specification is strongly influenced by a small number of design variables shall apply to a wide class of real-world MO problems. The rationale is henceforth. Goldberg suggests the need of some decomposability for being able to solve a single-objective optimization problem [11]. If all variables are epistatically linked, humans or an optimization algorithm doesn't stand a chance in finding a solution by an iterative process. In terms of MO problems, if all variables influence all objectives, a human iterative process cannot solve it, since the progress made in one step with respect to one specification will be overridden in the very next when a different specification is considered. Similarly a single variation step of a MO-EA will disturb all objectives. This leads us to believe that any problem where humans have met with some success need to have the aforesaid structure, i.e. sparse and partially non-overlapping influence of variables on objectives.[2] This gives a new notion of building-blocks for MO-EAs, where building-blocks could be considered as set of variables correlated with each specification. Variations in each of these building block would *influence most* a single objective, with some overlap with others. If the hypothesis here indeed holds, a new notion of 'hardness' of MO-EA problems can be based on the amount of epistasis between objectives. A direct measure of this epistasis is the number of correlated variables in two objectives which overlap. We can construct our MO-EAs test suites to reflect this structure and control 'hardness' on basis of number of variables influencing an objective and the overlap in their influence. This research can take a similar trajectory to that of GAs, i.e. to construct harder and harder problems and design efficient MO-EAs to solve them.

The second property, i.e. monotonicity of objectives with design variables comes as a bonus in COP. Not only do we know the variables correlated with each objective, we know the direction to vary them to improve the objective. Given this property, the designer doesn't require to do math after each iteration to find out which direction to move the variables. Will this property transfer to other problems? We don't know the answer to this yet. We know all linear problems have this property. Also, polynomials with all positive terms has this property. In [4], it is shown that many different real-world optimization problems yield such a structure. Also, on observation of test-suites traditionally used in MO-EAs, we found some of the sets have this property. These observations give us hope that some real-world problem do have this property. Even if they not, this can be incorporated as a 'measure of hardness' of the problem and efficient algorithms can be designed to tackle them.

4. COSMO

4.1 Description of COSMO

In the previous section we identified the set of variables correlated with each objective, which we loosely termed as 'building blocks' for the objective. We also derived the direction of variation for these variables for improving the objective.

We use this information to design a variation operator: COSMO which is implemented in the following simple way. For a given individual, randomly choose one of the objectives. Vary the point to *greedily* optimize the chosen objective in the local neighborhood. This is done by varying the variables correlated to the objective by a small amount in the direction to improve the objective. This greedy optimization is not done deterministically. The probability of variation of the correlated variables is increased in comparison to uncorrelated variables rather than effectively set to 1.0. This maintains the stochastic nature of the operator and respects the fact that the structural information (e.g. in Table 1), only describes first order effects.

COSMO optimizes (locally) an individual for one objective in one iteration, then for a different randomly-chosen objective in the next. This is probably at the expense of the prior objective in trade-off leading to interesting dynamics. This can be summarized as follows. COSMO pushes the points in direction of different objectives greedily (but locally) with dominated points getting periodically weeded out (by selection) leading to enumeration of the Pareto-optimal front. Stated another way, the algorithm can be summarized as greedy-walk across the search space of objectives coupled with periodic filtering to Pareto-optimal points through the selection phase.

COSMO's dynamics is similar to human-design process and we validate its effficiency from experiments. In the next section we provide a perspective on COSMO from the point of view of variation operators in EAs that explains why it could plausibly work.

4.2 A perspective on COSMO

To interpret COSMO, we begin by evaluating the kind of structure different operators exploit. As described in Section 2, operators use some structure in the selected solutions to create new solutions which are fitter. For instance, CMA-ES, which mimics the descent in Newton direction uses a quadratic assumption on the structure. EDAs on the other hand try to discover this structure online (constrained by expressibility of the chosen probability model). The bare minimum structural assumption an EA makes is that of locality: that fitter solutions lie *in the spatial vicinity* of already discovered fit solutions. In absence of this assumption, any search algorithm reduces to being a monte-carlo search. The assumption of locality is exploited in all EAs by controlling step size or equivalently the variance of the

[2] If the problem is not decomposable in this manner, the human needs to position themselves in a part of search space, where this structure holds to make any headway towards a solution.

Spec		Iref	Ir1	Ir2	V1	V2	V3	Rc	Cc	L1	L7	L3	L5	No. of Var.
gain	Corr	0	0	0	1	1	0	0	0	1	1	1	1	6
	Dir	0	0	0	-1	-1	0	0	0	1	1	1	1	
w_u	Corr	0	1	0	0	1	0	0	1	2	2	2	2	2
	Dir	0	1	0	0	-1	0	0	-1	-1	-1	-1	-1	
PM	Corr	0	1	1	1	1	0	0	1	0	0	0	0	5
	Dir	0	-1	1	-1	1	0	0	1	0	0	0	0	
Power	Corr	1	1	1	0	0	0	0	0	0	0	0	0	3
	Dir	1	1	1	0	0	0	0	0	0	0	0	0	
Area	Corr	2	2	1	1	2	2	2	1	2	2	2	2	3
	Dir	1	1	-1	1	-1	1	0	1	1	1	1	1	
SR	Corr	0	1	1	0	0	0	0	1	0	0	0	0	3
	Dir	0	1	1	0	0	0	0	-1	0	0	0	0	

Table 1: Summary of structural Information. No. of var. refers to the number of strongly correlated variables

sampling model. In Particle Swarm optimization, the use of locality is more obvious, where the new solutions explore in *vicinity* to the global and local best.

In low dimensional spaces, the simplest way to exploit locality is to modify all variables by a small amount. As the solution dimensions increase, however, a much smaller variation step (scaled by the square root of the number of dimensions) is required to maintain locality. Also, if the objective function is misbehaved with respect to even one of the dimensions, varying all dimensions isn't a good idea. A way to resolve this is by exploiting what we term as *dimensional locality*. Only vary variables on a small number of dimensions at each variation step. The structural assumption here is that within the hypervolume that is defined by fixing a few dimensions of a fit point there exists higher fitness points. This structural assumption is fairly general and a wide set of real-world problems have it. Additionally, a schedule can be used as to how many variables should be kept fixed, leading the search from being explorative initially to exploitative towards the end.

We can now interpret the SBX in light of dimensional locality. SBX implements binary crossover with each variable it acts upon. Deb recommends to use the crossover for half of the variables. If we assume, for a first order analysis, that SBX-crossover is effectively randomly mutating the variables it acts upon, SBX can be interpreted as exploiting *dimensional locality* for its success. In [9], it is shown that SBX inherently implements a self-adaptive schedule for controlling variance of the mutated variables.

To effectively exploit dimensional locality in an EA, there are potential roadblocks. First, in a high dimensional problem, only a few variables would be strongly correlated with the objective function (depending on the position of the individual). How can the appropriate small set of correlated variables be identified? Second, even if correlated variables are determined, the direction in which to vary them may not be known. The two above effects combined shall lead to small improvement in a single mutation step and thus make the search slower. Due to dimensional locality of the search, this can also lead to premature (local) convergence if the selection pressure is high.

These roadblocks do not necessarily require explicit structural information from the domain knowledge. Potentially, the correlation and sensitivity information could be generally observed and estimated by recording fitness changes before and after variations as long as a weak assumption of invariance with respect to other correlations is observed. In the case of circuits, the explicit correlation information for each objective is available from the first order circuit equations. As a bonus, the equations also signal the global direction of variation for each variable.

COSMO essentially exploits dimensional locality. It uses the available correlation and sensitivity information to address the roadblocks associated with exploitation of dimensional locality. This makes the search faster and more efficient as compared to a null operator exploiting dimensional locality. This has been tested in the experiments where we compare COSMO with both SBX and a NULL operator.

Note that one application of variation via COSMO does not generate a 'fitter offspring' in the sense of Pareto-optimality or in terms of dominating the parent. But successive application of COSMO will lead faster to Pareto optimal solutions. The elegance of COSMO lies in its simplicity (of concept and implementation) and the huge pay-back it gives. It is not the only way the structure identified in MO-EA problems can be exploited, but is one of the simplest way.

4.3 COSMO: Simple Extensions

COSMO can be applied to other MO-EA problems other than circuits as well. If equations for the fitness function are available, the correlation and sensitivity information can be derived in a similar way as done for circuits. In the case that the objectives are not monotonic with variables, a modified version of COSMO would vary the variables correlated to the chosen objective by a small amount in a randomly chosen direction.

If the equations are not available, this information can be extracted from a domain expert. Another way could be to do a sensitivity analysis on the objectives prior to execution of the algorithm. As mentioned in the previous section, the information extraction could also be done during the execution of the algorithm by recording fitness changes before and after variations. It will be interesting to see whether COSMO can improve the efficiency of optimization in other problem domains as well.

5. EXPERIMENTS

5.1 COSMO in NSGA II

We modify NSGA-II to include COSMO. NSGA-II uses

SBX. Let the fraction of design variables varied in a solution by SBX is λ (prescribed $\lambda = 0.5$ in [9]). For implementing COSMO, we replace this variation strategy by structural variation whose pseudo-code is given below.

```
foreach individual
  if violates constraint
    Find worst constraint violation spec ($spec_c$)
    Apply COSMO for improving $spec_c$
  else
    Choose any objective
    Apply COSMO for improving objective
end
```

If the solution violates a constraint, it is varied to improve the specification of maximum violation (rather than being optimized for any objective).

For COSMO, the number of design variables varied is on average $\alpha_1 = \lambda * n$, where n is the number of design variables. We define α_2 as the average number of correlated variables that are varied, which implies $(\alpha_1 - \alpha_2)$ is the average number of uncorrelated variables that are varied. All correlated variables have equal probability of variation and same is true for all uncorrelated variables. For a variable with no directional information, a gaussian random number, $N(0, \beta)$ is added to it. If the variable has directional information, the information is consulted to add a one-sided gaussian random number in the appropriate direction. Weakly correlated and uncorrelated variables are treated in the same way.

To create a null hypothesis for COSMO, we devise a NULL operator. The NULL operator varies the same average number of design variables, α_1, however chooses the variables to vary and their direction randomly without any structural information. This operator exploits dimensional locality, but uses no additional knowledge.

Linear Constraint Management: We use a DC point formulation for circuit design variables [18]. The design variables are constrained to lie in a polyhedron (given by equations) for circuit operatability (above-threshold and saturation constraint). Any circuit with design variables lying outside the polyhedron is unacceptable and its fitness is 0. Therefore we *constrain* the circuit design variables to remain inside the polyhedron. This results in a dynamic range for any variable depending on value of other variables. To tackle this, the range of all variables is determined according to the current value of other variables, every time variation to any variable is applied. If any variables falls out of a calculated range on variation, it is set to the min/max accordingly. We do not apply dynamic constraints to simple opamp, but apply them to FC-CS opamp, in which case they are critical.

This results in three algorithms: NSGA-SBX (SBX with crossover probability p_c and polynomial mutation), NSGA-NULL and NSGA-COSMO.

Genetic Algorithm Parameters: Four different algorithms were implemented as listed in Table 2. The population size is 100, tournament size is 2, maximum generations was set to 1000 and 10 runs were done for each algorithm. The variance, β was set a one-fourth of the range of the design variable. For fair comparison, NSGA-SBXa replicates NSGA-II parameter values [8], while $NSGA - SBXb$ matches the variation probability of NSGA-NULL and $NSGA - COSMO$.

The value of tournament size and crossover probability were set as prescribed in [8]. The value of α_i was informed by the average number of correlated variables (3.6). They were set intuitively without too much experimentation and there is probably scope for improvement. Similarly the value of β was set intuitively and performance could be improved by implementing a fixed schedule to vary the variance or by using self-adaptation.

Alg-Name	Parameters
NSGA-SBXa	$\lambda = 0.5$, $p_c = 0.9$
NSGA-SBXb	$\lambda = 0.25$, $p_c = 0.9$
NSGA-NULL	$\alpha_1 = 3$
NSGA-COSMO	$\alpha_1 = 3$, $\alpha_2 = 2.5$

Table 2: GA parameters used in the experiments.

5.2 Problem Description

Experiments were conducted for the simple opamp of Figure 1 and also an FC-CS opamp. The operating voltage was set to 1.8V and Silicon technology is TSMC 0.18u. The simple opamp is a 12 dimensional optimization problem, while the fully-cascoded opamp is a 20 variable optimization problem. The four objectives of optimization considered were gain, f_u (unity-gain frequency), power and area. Constraints were enforced on gain (> 1000), phase margin (> 40) and f_u ($> 5MHz$).

5.3 Results

The true Pareto-front is not known for COP. We do pairwise comparison between algorithms using the metric of coverage statistics(CS) [6]. For clarity, we report $DS = (1 - CS)$. For any two given set of Pareto-optimal points, Set a and b, DS_{ab} (Dominated Statistics) is equal to the fraction of points in Set a dominated by (inferior to) points in Set b and DS_{ba} implies exactly the opposite. [3].

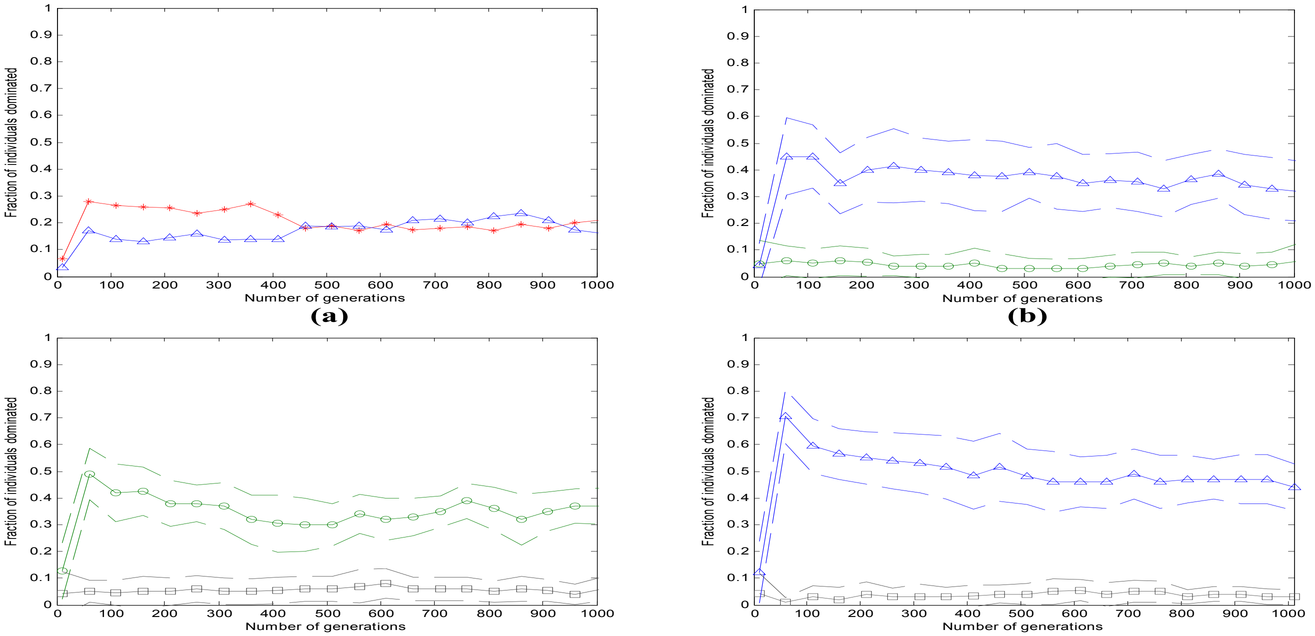
We take pairs of algorithms and calculate DS for the Pareto-optimal sets generated every 50 generations (See Figure 2). Each plot shows the median of the DS of 100 ordered pairs resulting from 10 runs of each of the two algorithms We also illustrate the standard deviation in the Figure.

Figure 2a shows the median value of DS for NSGA-SBXa and $NSGA - SBXb$. NSGA-SBXa shows slightly better performance initially and we use it for further comparisons. Figure 2b shows the comparison of NSGA-SBXa with NSGA-NULL. Around 5% of solutions are dominated for NSGA-NULL, while more than 40% points for NSGA-SBXa are dominated. Clearly, NSGA-NULL outperforms NSGA-SBXa.

We then compare NSGA-NULL and NSGA-COSMO (Figure 2c). Less than 5% of points for NSGA-COSMO are dominated, while more than 30% points for NSGA-NULL are dominated. If one compares the algorithms at 200 generations, the gain for $NSGA - COSMO$ is even higher, DS being 38% for NSGA-NULL and 5% for NSGA-COSMO. In Figure 2d we also show the comparison of NSGA-SBXa and NSGA-COSMO. This establishes the superiority of COSMO to SBX and NULL operator.

To study the speed of convergence of NSGA-COSMO, we compared NSGA-NULL's Pareto-set at each generation with NSGA-COSMO's Pareto-set created at 100th generation. We found that even at 1000 generations NULL does not perform as well as COSMO at the 100th generation.

[3] DS_{ab} and DS_{ba} do not add to 1 since some points in both sets will be Pareto-optimal and are not counted in either statistic



(a) **(b)** **(c)** **(d)**

Figure 2: Pairwise domination comparison of the Pareto-optimal sets of the 4 test algorithms. A lower fraction of individuals dominates implies superiority. See text for details of the DS coverage metric. NSGA-SBXa is △, NSGA-SBXb is + ,NSGA-NULL is o and NSGA-COSMO is □. Results are compiled for 10 runs. Solid line with symbol is median, dashed lines are range of standard deviation. Plot a compares NSGA-SBXa and NSGA-SBXb, plot b: NSGA-SBXa and NSGA-NULL, plot c: NSGA-NULL and NSGA-NULL and plot d: NSGA-SBXa and NSGA-COSMO.

At any time more than 25% solutions for NSGA-NULL are dominated, while not more than 12% of solutions for NSGA-COSMO at 100th generation are dominated. This clearly shows that SMO gives faster and more efficient optimization.

DS statistics doesn't show how much better the solutions are numerically between NSGA-COSMO and NSGA-NULL. On studying some of the results manually we found that NSGA-COSMO's non-dominated solutions were significantly better as compared to that of NSGA-NULL, while NSGA-NULL's non-dominating solutions were only slightly better. For instance, one of the solution for NSGA-COSMO at 1000th generation had slightly better gain, double unity gain frequency, half the area and one-third the power of the dominated solution of NSGA-NULL. From the view point of an analog designer, this is a large improvement.

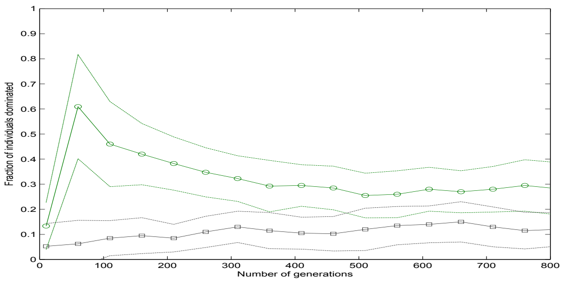


Figure 3: Comparison of NSGA-NULL and NSGA-COSMO for FC-CS opamp

We also gathered preliminary results for an FC-CS opamp. The comparison between NSGA-NULL and NSGA-COSMO is shown in Figure 3. At the 100th (200th) generation, only 8.5% (8.5%) solutions on average are dominated for NSGA-COSMO, while 46% (38.25%) of solutions of NSGA-NULL are dominated. The performance at a lower number of generations is important to study, since for larger designs the longer time to simulate leads to less generations being executable in a realistic time-frame. At the 800th generation, NSGA-COSMO has 29% solutions dominated, while NSGA-COSMO has only 11.5% solutions dominated.

Discussion: We find that the NULL operator performs better than SBX. The only difference in the two operators is the way they handle the variance of the mutated variables [9]. This could be studied further to explain this difference.[4] This indicates that the complex process of mutation of SBX is of no value for the given problem and straightforward exploitation of *dimensional locality* is sufficient. The higher accuracy of optimization by COSMO as compared to NULL operator is evidence to the premature convergence of operators exploiting dimensional locality. The faster convergence of NSGA-COSMO is also valuable, specially for domains like that of circuits, which have a high fitness evaluation time.

6. CONCLUSIONS

We identify the structure in circuit problems and discuss how it is indicative of general structure and measure of hardness in MO-EA problems. We then present a methodology

[4] In Section 4, we conjectured that SBX crossover on a variable can be considered mutation by first order analysis

to exploit this structure in a MO-EA by designing a novel correlation sensitive operator, COSMO. COSMO is, at the least, universally applicable in the domain of circuits and can be extended to other domains. We finally compare COSMO's performance with that of the traditional SBX (with polynomial mutation) and a NULL operator. For two instances of circuits, we show that COSMO gives faster and better optimization than both these operators. The paper also makes contribution toward interpretation of operators used for MO-EAs.

7. FUTURE WORK

There are a few different directions for future work, which have been expressed in different sections of the paper. Here, we summarize a few meta-themes.

1. Structural analysis of real-world MO optimization problems in different domains to observe if they have the structure exhibited by COP. If they do, COSMO can be used for their efficient optimization.

2. Artificial construction of problem exhibiting the hypothesized structure of MO-EA problems and design of efficient algorithms to solve the problems.

3. In context of COSMO, further improvements and refinements could be made. A self-adaptive or fixed schedule of variance could be implemented. Also, studies needs to be conducted on how many variables to vary for efficient optimization and if this should be adaptive.

4. COSMO is just one way of exploiting the structural information. An extension could be greedily optimize for more than one objective in a variation step and design operators to this end. The holy-grail remains to design operators which will produce an offspring that dominates the parent. For designing such operators, one needs not only the sensitivity direction, but also its value. This sensitivity value can be made available from equations in COP. It is a promising research direction to incorporate this information to design operators which produce dominating offsprings.

8. REFERENCES

[1] V. Aggarwal and U.-M. O'Reilly. Monotonicity information in Simulation-based approaches for efficient circuit sizing and hierarchical synthesis. In *In submission to ICCAD*, 2007.

[2] T. Baeck and H.-P. Schwefel. An overview of evolutionary algorithms for parameter optimization. *Evol. Comput.*, 1(1):1–23, 1993.

[3] P. Bosman and D. Thierens. Numerical optimization with real-valued estimation-of-distribution algorithms. *Scalable Optimization via Probabilistic Modeling: From Algorithms to Applications*, pages 91–120, 2006.

[4] S. Boyd, S.-J. Kim, L. Vaudenberghe, and A. Hassibi. A tutorial on geometric programming. In *Technical report, EE Department, Stanford University*, 2004.

[5] M. Chu, D. J. Allstot, J. M. Huard, and K. Y. Wong. NSGA-based parasitic-aware optimization of a 5GHz low-noise VCO. *ASP-DAC*, 00:169–173, 2004.

[6] K. Deb. *Multi-Objective Optimization Using Evolutionary Algorithms.* John Wiley & Sons, Inc., New York, NY, USA, 2001.

[7] K. Deb and R. B. Agrawal. Simulated binary crossover for continuous search space. *Complex Systems*, 9:115–148, 1995.

[8] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimisation: NSGA-II. In *PPSN*, pages 849–858, 2000.

[9] K. Deb and H. Beyer. Self-adaptive genetic algorithms with simulated binary crossover. Technical Report No. CI-61/99, University of Dortmund, Department of Computer Science/LS11, 1999.

[10] T. Eeckelaert, T. McConaghy, and G. Gielen. Efficient multiobjective synthesis of analog circuits using hierarchical pareto-optimal performance hypersurfaces. In *DATE '05*, pages 1070–1075, 2005.

[11] D. E. Goldberg. *The Design of Innovation: Lessons from and for Competent Genetic Algorithms.* Kluwer Academic Publishers, Norwell, MA, USA, 2002.

[12] N. Hansen, S. Muller, and P. Koumoutsakos. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation. *Evolutionary Computation 11 (2003) 1–18*, 2003.

[13] J. R. Hellums, L. R. Carley, M. J. Krasnicki, R. A. Rutenbar, and R. Phelps. A case study of synthesis for industrial-scale analog IP: Redesign of the equalizer/filter frontend for an ADSL CODEC. *DAC*, 00:1–6, 2000.

[14] C. Igel, N. Hansen, and S. Roth. Covariance matrix adaptation for multi-objective optimization. *to appear in Evolutionary Computation*, 2007.

[15] Y. Jin and B. Sendhoff. Connectedness, regularity and the success of local search in evolutionary multi-objective optimization. In *Proceedings of IEEE Congress on Evol. Comput.*, pages 1910–1917, 2003.

[16] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of the Fourth IEEE International Conference on Neural Networks.*

[17] J. D. Knowles and D. W. Corne. Approximating the nondominated front using the pareto archived evolution strategy. *Evol. Comput.*, 8(2):149–172, 2000.

[18] F. Leyn, G. G. E. Gielen, and W. M. C. Sansen. An efficient DC root solving algorithm with guaranteed convergence for analog integrated CMOS circuits. In *ICCAD*, pages 304–307, 1998.

[19] M. Pelikan, K. Sastry, and E. C. Paz. *Scalable Optimization via Probabilistic Modeling: From Algorithms to Applications.* Springer-Verlag, Berlin Heidelberg, 2006.

[20] S. K. Tiwary, P. K. Tiwary, and R. A. Rutenbar. Generation of yield-aware pareto surfaces for hierarchical circuit design space exploration. In *DAC*, pages 31–36, 2006.

[21] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. In *EUROGEN 2001, Evolutionary Methods for Design, Optimization and Control with Application to Industrial Problems*, pages 95–100, Greece, 2002.