Multi-objective Gene-pool Optimal Mixing Evolutionary Algorithms[1]

Ngoc Hoang Luong Han La Poutré Peter A.N. Bosman

Centrum Wiskunde & Informatica (CWI), Amsterdam, The Netherlands

1 Introduction

Multi-objective (MO) optimization problems deal with multiple conflicting objectives at the same time. Each solution represents a trade-off between different objectives, and a utopian solution optimizing all objectives is non-existent. Instead of searching for one best solution as in single-objective (SO) optimization, MO optimizers try to obtain a good approximation $\mathcal{S}$ of the true Pareto-optimal front $\mathcal{P}_F$ of the problem at hand, which consists of multiple non-dominated solutions. The goal of MO optimization has thus two aspects: proximity (i.e. the approximation set $\mathcal{S}$ should be as close as possible to $\mathcal{P}_F$) and diversity (i.e. set $\mathcal{S}$ should be well-spread along $\mathcal{P}_F$). It has been showed that MO evolutionary algorithms (MOEAs) are well-suited and, indeed, an effective methodology for solving MO optimization problems. However, while classical MOEAs are effective in achieving this two-fold goal, the issue of scalability, which depends on the algorithms' capability for linkage relation learning, is often overlooked. MO estimation-of-distribution algorithms (MOEDAs), such as the Multi-objective Adapted Maximum-Likelihood Model (MAMaLGaM) or the Multi-objective Hierarchical Bayesian Optimization Algorithm (mohBOA), tackle the scalability issue by replacing classic variations operators with model-based variation operators.

In this paper, by constructing the Multi-objective Gene-pool Optimal Mixing Evolutionary Algorithm (MO-GOMEA), we pinpoint key features for scalable MO optimizers. First, an elitist archive is beneficial for keeping track of non-dominated solutions. Second, clustering can be crucial if different parts of the Pareto-optimal front $\mathcal{P}_F$ need to be handled separately. Next, an efficient linkage learning procedure with a lean linkage model is required to capture the underlying dependencies among decision variables. It is also important that the optimizers can effectively exploit the learned linkage relations to generate new offspring solutions, steering the search toward promising regions in the search space.

2 Key Features

Elitist Archive[1]. The number of Pareto-optimal solutions can be numerous or even infinite, exceeding the limited capacity of the population. Furthermore, some good solutions can be lost due to the stochastic nature of the selection and variation operators. It is thus beneficial to have an archive keeping track of the non-dominated solutions found so far, and, in turn, the current Pareto front constituted by these solutions. The criteria for accepting a newly generated solution $\boldsymbol{x}$ into the archive $\mathcal{A}$ are that $\boldsymbol{x}$ is not dominated by any existing members of $\mathcal{A}$ and that $\boldsymbol{x}$ can change the Pareto front found so far. If $\boldsymbol{x}$ can be added to $\mathcal{A}$, solutions dominated by $\boldsymbol{x}$ are removed from $\mathcal{A}$. Mechanisms to maintain the diversity or to adapt the archive when its capacity is exceeded can also be employed.

Clustering [1]. In every generation, the population of candidate solutions is partitioned into k equal-sized clusters. Each cluster can be seen as being allocated an equal amount of resource for approaching a different part of the Pareto-optimal front $\mathcal{P}_F$. The extreme solutions (i.e. the one that is optimal in terms of a single objective) and their corresponding extreme regions on $\mathcal{P}_F$ can be very inefficient

[1]The full paper has been accepted for presentation at *The Genetic and Evolutionary Computation Conference (GECCO 2014)*.

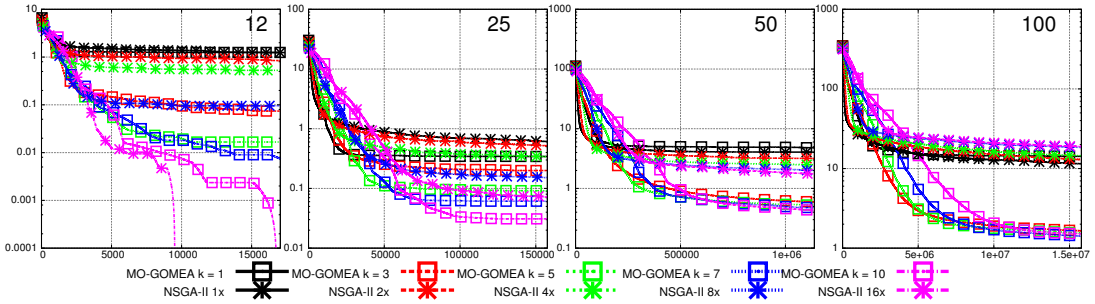

Figure 1: Performance of MO-GOMEA and NSGA-II on multi-objective MAXCUT problems of different problem sizes. Horizontal axis: number of evaluations. Vertical axis: $\boldsymbol{D}_{\boldsymbol{\mathcal{P}}_{\boldsymbol{F}} \to \mathcal{S}}$

for MO optimization to obtain. It is thus more beneficial if these extreme regions are handled by SO optimization. Therefore, for every objective, the cluster having the best cluster mean value in that objective is assigned to perform SO optimization operations regarding that objective.

Linkage Learning [2]. After partitioning the population, a set $\boldsymbol{\mathcal{S}}_i$ of solutions having better fitness values is selected from each cluster $\boldsymbol{\mathcal{C}}_i$. A linkage model $\mathcal{F}_i$ for cluster $\boldsymbol{\mathcal{C}}_i$ is then learned from this selection set $\boldsymbol{\mathcal{S}}_i$. MO-GOMEA employs the Linkage Tree (LT) structure as its linkage model. The LT contains linkage groups representing the dependencies among decision variables, and organizes them in a tree-like fashion. Leaf nodes of LT are univariate linkage groups, capturing decision variables as being fully independent. Branch nodes of LT are multivariate linkage groups, indicating that the decision variables in each group have some degree of dependency and thus should be treated jointly together when performing variation. The LT can be learned by an efficient hierarchical clustering procedure.

Optimal Mixing (OM) [2]. From each existing solution $\boldsymbol{x}$, the OM operator uses the LT $\mathcal{F}_i$ of the cluster $\boldsymbol{\mathcal{C}}_i$, to which $\boldsymbol{x}$ belongs, to construct a new offspring solution $\boldsymbol{o}$ by iteratively improving $\boldsymbol{x}$. First, $\boldsymbol{o}$ is cloned directly from $\boldsymbol{x}$. Looping through the LT $\mathcal{F}$, for every linkage group $\boldsymbol{F} \in \mathcal{F}$, a random donor solution $\boldsymbol{p}$ is selected from the same cluster $\boldsymbol{\mathcal{C}}_i$. The values of the decision variables indicated by $\boldsymbol{F}$ are copied from $\boldsymbol{p}$ to $\boldsymbol{o}$. If such mixing create a new (partial) solution $\boldsymbol{o}'$ which dominates $\boldsymbol{o}$ or is a non-dominated solution regarding the archive $\boldsymbol{\mathcal{A}}$, then $\boldsymbol{o}'$ is accepted as the new $\boldsymbol{o}$. Otherwise, $\boldsymbol{o}$ is rolled back to the previous state. Note that for solutions $\boldsymbol{x}$ belonging to extreme clusters, OM improves $\boldsymbol{x}$ in SO optimization manner.

3 Experiment & Discussion

We show the performance of our MO-GOMEA with the well-known NSGA-II on solving the MO-MAXCUT. Results of other benchmarks can be found in the full paper. We experiment with instances of MO-GOMEA having different number of cluster $k = 1, 3, 5, 7, 10$. Multiple instances of NSGA-II are created by scaling the base population size to 1,2,4,8, and 16 times. We employ the inverse generational distance $\boldsymbol{D}_{\boldsymbol{\mathcal{P}}_{\boldsymbol{F}} \to \mathcal{S}}$ as the performance indicator. Fig. 1 shows convergence graphs of $\boldsymbol{D}_{\boldsymbol{\mathcal{P}}_{\boldsymbol{F}} \to \mathcal{S}}$ from the beginning until termination, averaged over 100 independent runs. For MAXCUT instance having problem size $l = 12$, only NSGA-II $16\times$, the one with the largest population size, has better performance than MO-GOMEA. However, as the problem size increases ($l \geq 25$), MO-GOMEA starts to outperform NSGA-II. The bigger the problem is, the wider the performance gap between MO-GOMEA and NSGA-II becomes, indicating the superior scalability of MO-GOMEA over NSGA-II. Note that MO-GOMEA without clustering ($k = 1$) has performance results relatively the same to those of NSGA-II instances, which do not have clustering either. This emphasizes the importance of clustering to handle different parts of the Pareto-optimal front separately. Without linkage learning, NSGA-II cannot solve problems of large problem sizes as efficiently as MO-GOMEA even though it is equipped with very large population sizes (NSGA-II $8\times$ or $16\times$). Having a linkage learning mechanism and effectively exploiting the linkage model are thus crucial for a scalable MO optimizer.

References

[1] P. A. N. Bosman. The anticipated mean shift and cluster registration in mixture-based edas for multi-objective optimization. In *Proc. of the Genetic and Evolutionary Comp. Conf. - GECCO-2010*, pages 351–358.

[2] P. A. N. Bosman and D. Thierens. More concise and robust linkage learning by filtering and combining linkage hierarchies. In *Proc. of the Genetic and Evolutionary Comp. Conf. - GECCO-2013*, pages 359–366.