Co-Evolutionary Approaches for Numerical Optimization

Min-Jea Tahk

Division of Aerospace Engineering, KAIST, Daejeon 305-701, KOREA

Corresponding Author: Min-Jea Tahk, mjtahk@fdcl.kaist.ac.kr

ABSTRACT

Co-evolution is a powerful tool to solve numerical optimization problems when conventional methods are not much useful. By properly selecting the fitness measure (or selection strategy) for each population group, various optimization problems can be handled without much difficulty. As an overview of the recent developments in the co-evolution methodology, this paper summarizes the co-evolution methods developed for constrained optimization, robust control design, and dynamic games. The scope of this paper is limited to the works the author has been involved with.

MATRIX GAMES AND CO-EVOLUTION

<u>Co-evolution for Saddle-Point Problems</u>

Consider a static zero-sum game G for which a payoff function $f = f(u,v)$ is to be minimized by $u \in \mathbb{R}^p$ and maximized by $v \in \mathbb{R}^q$. If G has a solution of pure strategy, then

$$\min_u \max_v f(u,v) = \max_v \min_u f(u,v) \quad (1)$$

and this solution is called the saddle point solution. Somewhat surprisingly, saddle-point problems over the domain $\mathbb{R}^p \times \mathbb{R}^q$ are quite difficult to solve by using a gradient-based technique. Contrary to minimization (or maximization) problems, these problems do not have solutions on a geometrically stable region. Iteration of minimization and maximization may lead to a solution but may diverge in some cases.

Co-evolution, which uses evolution of two population groups, can be used to solve saddle-point problems. The basic idea is to evolve a matrix game to the region of the solution. The matrix game is defined by the individuals of the two opposing groups at each generation while each group evolves by using its own security strategy.

The procedure of co-evolution for solving saddle-point problems is sketched as follows:

1) Initialize the population groups U and V: Individuals may have additional parameters such as variance size and directionality.
2) From the matrix game defined by U and V, calculate

$$\overline{r}(u^i) = \max_v f(u^i, v) \quad v \in V \quad (2)$$

$$\underline{r}(v^j) = \min_u f(u, v^j) \quad u \in U \quad (3)$$

for all individuals of U and V: Here u^i and v^j denote the object parameter values of the individuals. Then, the values of $\overline{r}(u^i)$ and $\underline{r}(v^j)$ determines the fitness of u^i and v^j based on security strategy.

3) Apply an evolutionary method for selection and recombination (or reproduction) to produce the population of the next generation: A genetic algorithm or the evolution strategy can be used for this purpose.
4) Go back to Step 2) if convergence is not achieved.

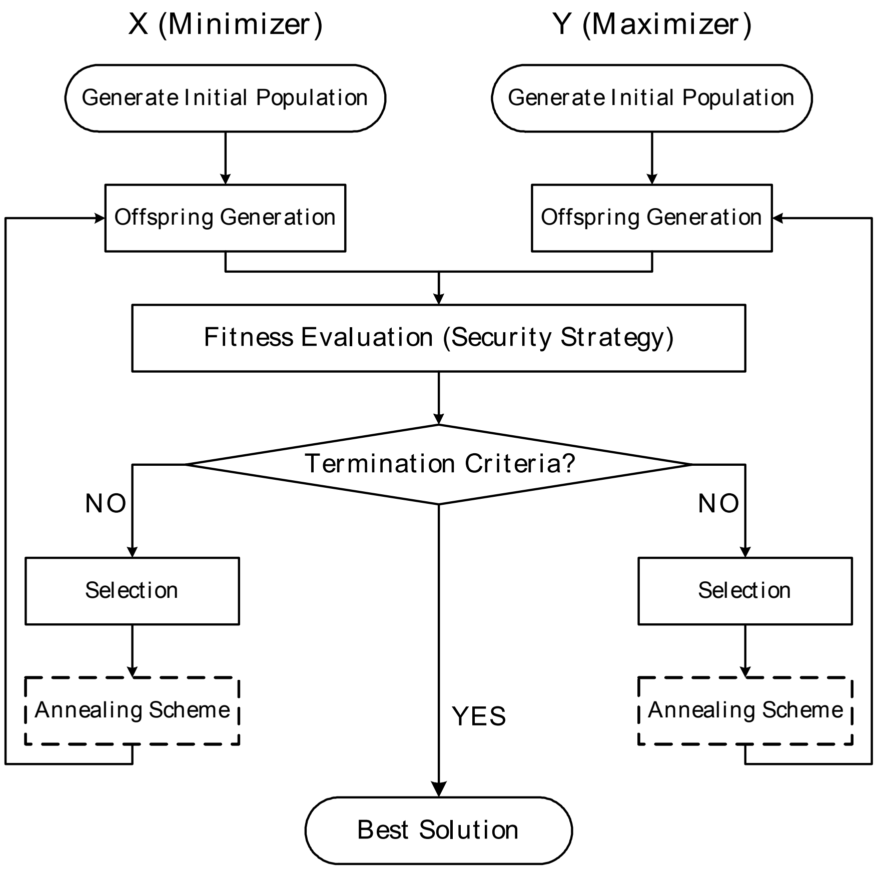

Fig. 1 Flow chart of the conventional co-evolutionary algorithm

Fig. 1 shows the flowchart of the proposed co-evolutionary algorithm. Except that a fitness evaluation procedure based on security strategy is used, a co-evolutionary process is basically a parallel evolution process of two population groups.

<u>Co-evolution for Asymmetric Games</u>

Usual min-max or max-min problems are different from saddle-point problems in that there is a leader and a follower. In general, there is no guarantee that such problems have the saddle point for the co-evolutionary process described above to work properly. Consider a min-max problem given by

$$\min_u \max_v f(u,v) \qquad (4)$$

where we assume that there does not exist the saddle point. In (4), u is the leader and v is the follower. For any u, we can find the optimal v^*, which is a function of u. Hence, we can transform (4) to a minimization problem given by

$$\min_{u} f(u, v^*(u)) \qquad (5)$$

A straightforward application of an optimization method will result in a two-loop optimization scheme; the outer loop is for u and the inner loop is for v. Observe that this scheme is not suitable for evolutionary methods since each generation of the outer loop requires a complete evolution process of the inner loop.

To maintain the parallel structure of the co-evolutionary approach used for saddle-point problems, the fitness measure for the follower should be based on the Stackelberg solution rather than the security strategy; the fitness of individuals of the follower population should be determined not to optimize the worst-case outcome but with respect to the best individual of the leader population. For this purpose, a method for fitness evaluation called "man-to-man strategy" has been developed. Use of an additional random population in V (or use of heterogeneous population) also greatly improves the convergence characteristics

APPLICATIONS

Constrained Optimization (Augmented Lagrangian Methods)

The most successful application of co-evolution for numerical optimization is Co-Evolutionary Augmented Lagrangian Method(CEALM) proposed in [Tahk98, Tahk00]. By using the augmented Lagrangian formulation, a constrained optimization problem can be transformed to a saddle-point problem between the parameter vector x and the multiplier vectors. Consider a general constrained optimization problem

$$\min_{x} f(x) \qquad x \in \mathbb{R}^n \qquad (6)$$

subject to

$$g_i(x) \le 0, \quad h_i(x) = 0, \quad L_i \le x_i \le U_i \qquad (7)$$

If the primal problem is convex over S (f and g are convex and h is affine over S), then the Lagrangian has the saddle point at (x^*, μ^*, λ^*). For nonconvex problems, the augmented Lagrangian methods avoid this difficulty by convexifying f with quadratic penalty terms associated with the constraints.

The main issue of the augmented Lagrangian methods is how to update the Lagrange multiplier to get convergence to (μ^*, λ^*). Contrary to the previous approaches, CEALM takes a genuinely evolutionary approach, relying on the evolution of (μ, λ) as well as that of x to achieve the saddle point (x^*, μ^*, λ^*). The basic idea of CEALM is very simple: the saddle point of the augmented Lagrangian is found by the co-evolutionary method designed for saddle-point problems.

The pay-off function, which is the augmented Lagrangian for constrained optimization, should be evaluated $n_U \times n_V$ times for fitness evaluation. However, we only need to evaluate $f(x)$, $g(x)$, and $h(x)$ for n_U times; only once for each x^i. Once $f(x)$, $g(x)$, and $h(x)$ are evaluated for an x^i, the value of $L_A(x^i, \mu^j, \lambda^j, \rho)$ for any j can be obtained by a simple calculation. Therefore, the computational burden of the co-evolutionary method is comparable to that of the other evolutionary methods using a single group.

Parameter Robust Control

The co-evolution approach has been applied to missile autopilot design problems that are formulated as min-max problems to obtain controllers robust to aerodynamic uncertainties [Park97, Park98]. These works on minimax control design are extended in [Hur03] to find a robust controller satisfying the design requirements. Parameter robust control is a min-max problem given as

$$\min_{k \in K} \max_{p \in P} \quad f(k, p) \tag{8}$$

where k denotes controller parameters, p system parameter uncertainties, and K and P are the set of all possible k and p, respectively.

Robust control problems are different from constrained optimization problems in that there may not exist the saddle-point solution. Hence, we need a co-evolution algorithm which provides good stability even for the case of no saddle point. As previously noted, a heterogeneous population, in which one part takes the Stackelberg strategy and the other part takes the random strategy, is suitable for p, the follower. Another difference is that the cost function should be evaluated by $n_U \times n_V$ times, where U and V are the population representing K and P.

CONCLUSION

Experience with many practical applications shows that the co-evolution approach can solve most of numerical optimization problems successfully if the selection strategy (or fitness measure) is properly selected for each group involved in the matrix game derived from the given optimization problem. The most attractive feature of the co-evolution methods is their capability of solving various problems that the gradient-based methods or single-group evolution methods cannot handle easily. Although the co-evolution methods are not fast, future study is believed to widen the application area of co-evolution further.

REFERENCES

[Hur03] J.Hur, H.G.Lee, and M.J.Tahk, "Parameter robust control design using bimatrix coevolution algorithms," Engineering Optimization, Vol.35, No.4, pp.417-426, Aug. 2003.

[Park97] C.S.Park and M.J.Tahk, "A numerical solver based on evolutionary computation for minimax problems," JSASS Aircraft Symposium, Tokyo, Japan, Oct. 1997.

[Park98] C.S.Park and M.J.Tahk, "A coevolutionary minimax solver and its application to autopilot design," AIAA Conference on Guidance, Navigation, and Control, Boston, Mass., USA, Aug. 1998.

[Tahk98] M.J.Tahk, "Co-evolution for engineering optimization problems: minimax design and constrained optimization," Invited Paper, JSASS Aircraft Symposium, Yokosuka, Japan, Oct. 1998.

[Tahk00] M.J.Tahk and B.C.Sun, "Co-evolutionary augmented Lagrangian methods for constrained optimization," IEEE Transactions on Evolutionary Computation, Vol.4, No.2, pp.114-124, July 2000.