Attraction Based PSO with Sphere Search for Dynamic Constrained Multi-Objective Optimization Problems

Jingxuan Wei
School of Computer Science and Technology
Xidian University
Xi'an, China
wjxjingxuan@gmail.com

Mengjie Zhang
School of Engineering and Computer Science
Victoria University of Wellington
Wellington, New Zealand
Mengjie.Zhang@ecs.vuw.ac.nz

ABSTRACT

Developing efficient algorithms for dynamic constrained multi-objective optimization problems (DCMOPs) is very challenging. This paper describes an attraction based particle swarm optimization (PSO) algorithm with sphere search for such problems. A dynamic constrained multi-objective optimization problem is transformed into a series of static constrained multi-objective optimization problems by dividing the time period into several equal intervals. To speed up optimization process and reuse the information of Pareto optimal solutions obtained from previous time, a new method based on sphere search is proposed to generate the initial swarm for the next time interval. To deal with the transformed problem effectively, a new particle comparison strategy is proposed for handling constraints in the problem. A local search operator based on the concept of attraction is introduced for finding good search directions of the particles. The results show that the proposed algorithm can effectively track the varying Pareto fronts with time.

Categories and Subject Descriptors

G.1 [**Numerical Analysis**]: Nonlinear programming, constrained optimization

General Terms

Algorithms, experimentation

Keywords

Particle Swarm Optimization, Dynamic Multi-Objective Optimization, Constrained Optimization

1. INTRODUCTION

Many real world optimization problems exist in dynamic environments. A dynamic constrained multi-objective optimization problem can be described as follows:

$$\begin{cases} \min & f(x,t) = \{f_1(x,t), f_2(x,t), ..., f_m(x,t)\} \\ \text{s. t.} & g_j(x,t) \le 0, j = 1, \cdots, p \\ & x \in [L,U] \end{cases} \quad (1)$$

where $t \in [a,b]$ is time variable, $x = (x_1, x_2, ...x_n)$ is the decision vector, $g_j(x,t)(j = 1, 2, ..., p)$ are inequality constraints. All of these constraints depend on time variable t. $[L,U] = \{x = (x_1, x_2, ...x_n)|l_i \le x_i \le u_i, i = 1, 2, ..., n\}$ is the search space. Constraint violation at time t is defined as $\Phi(x,t) = \sum_{j=1}^{p} \max(0, g_j(x,t))$. A vector $\mu = (\mu_1, \mu_2, ...\mu_n)$ is said to dominate a vector $\nu = (\nu_1, \nu_2, ...\nu_n)$ (denoted as $\mu \prec \nu$) if: $\forall i \in \{1, 2, ...m\}, f_i(\mu) \le f_i(\nu) \bigwedge \exists j \in \{1, 2, ...m\}, f_j(\mu) < f_j(\nu)$. A solution x is called a Pareto optimal solution for problem (1) at a fixed time t if $\Phi(x,t) = 0$ and $\sim \exists \widetilde{x} \in [L,U]$ such that $\Phi(\widetilde{x},t) = 0$ and $\widetilde{x} \prec x$.

A few studies have been made to develop effective evolutionary computation algorithms for solving dynamic constrained multi-objective optimization (DCMOPs), such as dynamic NSGA-II [1] and dynamic MOPSO [2]. When PSO is used to solve the dynamic problems, there are several issues. First, PSO should be able to track the varying Pareto fronts rather than a repeated start of optimization process. Second, an effective constraint handling technique needs to be developed to avoid premature convergence. Third, a local search is needed to make the convergence faster.

To address the above issues, the goal of the paper is to develop a new PSO method for dynamic constrained multi-objective optimization when the environment continuously changes with time. We expect the proposed algorithm to effectively track the varying Pareto fronts and the Pareto optimal solutions obtained in each time period to be widely distributed along the true Pareto front.

2. SPHERE SEARCH METHOD

When objective functions and constraints change with time continuously, the DCMOPs can be approximated by a series of static multi-objective optimization problems (SMOPs) by taking the time interval fixed. To speed up the optimization process, a sphere search method is proposed to generate the swarm of the next time interval.

Algorithm 2.1 (Sphere search method)

Step 1. Define spheres

- Initial radius for sphere r: $\sqrt{2}$
- Centroid of a sphere: $\widetilde{x} = (\widetilde{x}_1, ...\widetilde{x}_n)$ (randomly choose 11 points from the set of the Pareto optimal solutions as the centroids).
- Number of samples in each sphere: $S = 5$

These parameter values are chosen based on an empirical search via initial experiments.

Step 2. Compute S uniformly distributed points in each sphere by using Eq.2. $x_j = (x_{j1}, x_{j2}, ...x_{jn}), j \in \{1, 2, ...S\}$

GECCO'11, July 12–16, 2011, Dublin, Ireland.
ACM 978-1-4503-0690-4/11/07.

denotes one of these points. α_{jn-1} ranges in $[0, 2\pi]$, $\alpha_{ji} \in [0, \pi], i = 1, 2, ...n-2, j = 1, ..., S$.

$$\begin{aligned}
&x_{j1} = \widetilde{x}_1 + r_{ns}\cos(\alpha_{j1})\\
&x_{j2} = \widetilde{x}_2 + r_{ns}\sin(\alpha_{j1})\cos(\alpha_{j2})\\
&x_{j3} = \widetilde{x}_3 + r_{ns}\sin(\alpha_{j1})\sin(\alpha_{j2})\cos(\alpha_{j3})\\
&...\\
&x_{jn-1} = \widetilde{x}_{n-1} + r_{ns}\sin(\alpha_{j1})...\sin(\alpha_{jn-2})\cos(\alpha_{jn-1})\\
&x_{jn} = \widetilde{x}_n + r_{ns}\sin(\alpha_{j1})...\sin(\alpha_{jn-2})\sin(\alpha_{jn-1})\\
&j = 1, 2, ...S
\end{aligned} \tag{2}$$

Totally, 55 points are generated based on sphere search. If the size of the swarm of the next time is 100, then the 55 solutions produced by the sphere search can be seen as a part of the swarm, and the other 45 points are randomly selected from the search space.

3. ATTRACTION BASED PSO ALGORITHM

In order to solve the transformed problem effectively, we propose two new techniques. We first introduce a new particle comparison strategy to keep some infeasible particles. A good infeasible particle is defined as follows: if the constraint violation of one infeasible particle is less than a previously defined threshold value ψ, the particle is said to be a good one. $\psi = \frac{1}{T}\sum_{i=1}^{popsize}\Phi(x_i, t)/popsize$, where T is a parameter, which increases from 0.4 to 0.8 with the increasing generation number.

Algorithm 3.1 (Comparison Strategy)

Step 1. If two particles a_1 and a_2 are feasible solutions, we select the one with the smaller rank value.

Step 2. If two particles are infeasible, we select the one with the smaller constraint violation.

Step 3. Suppose one particle is feasible and the other is infeasible, if the constraint violation of the infeasible solution is smaller than the threshold, the two particles are compared according to step 1; otherwise, we choose the feasible particle.

While the proposed strategy is expected to improve the diversity of PSO, it cannot guarantee PSO to generate good particles with smaller constraint violations and rank values. To deal with this issue, we design a new local search operator based on the concept of attraction of force. Suppose particle y is selected to undergo local search, other particles in the swarm exert a force $F_i(x_i, y)$ to y, which is defined below.

Total Force: Imagine that there is a force from x_i to y, which is defined as Eq.3.

$$\begin{aligned}
F_i(x_i, y) \quad = \quad & sign(\Delta)\frac{x_i - y}{\|x_i - y\|}\cdot\frac{C - \Phi(x_i, t)}{rank(x_i)}\\
\cdot \quad & \exp[-\frac{1}{Dist(f(x_i), f(y))}]
\end{aligned} \tag{3}$$

where $sign$ is the sign function, $C = 10000$, so that $C - \Phi(x, t) > 0$. $rank$ denotes the rank value of a particle. $dist$ denotes the Euclidean distance. For a swarm pop include y, the swarm has a total force to y,which is defined as $F(y) = \sum_{x_i \in pop, x_i \neq y} F_i(x_i, y)$. The local search to y is a variation along the search direction $\frac{F(y)}{\|F(y)\|}$, and the new generated particle is defined as $fy_1 = y + \lambda \cdot \frac{F(y)}{\|F(y)\|}$, where $\lambda \in (0, 1)$ is the step size. Searching along this direction, better particles are expected to be found.

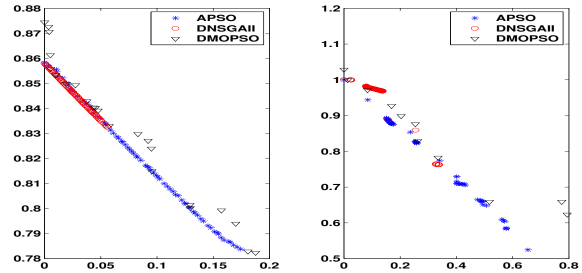

Figure 1: Evolved Pareto fronts: for DCTP1 at $t = 0.2$ (left); for DCTP2 at $t = 0.8$ (right).

3.1 Experiments and Comparisons

We choose two DCMOPs with complicated Pareto sets from [3], named DCTP1 and DCTP2. The performance of the proposed APSO is compared with DNSGAII [1] and DMOPSO [2]. Due to the page limit, we only depict the Pareto fronts found by the three algorithms at $t = 0.2$ for DCTP1, $t = 0.8$ for DCTP2. For the proposed APSO, we set a swarm size of 100, the local search probability 0.3, the maximal generation number 150 for each fixed time interval, $c_1 = c_2 = 2.0$; and the inertia weight $\omega = 0.4$.

From Fig.1, we can see that the APSO generally performs better than both DNSGAII and DMOPSO on all test problems, in both the convergence and diversity metrics.

4. CONCLUSIONS

The goal of this paper was to investigate a PSO based algorithm for DMOPs with constraints. The goal was successfully achieved by developing a new sphere search method, a new comparison strategy and an attraction based local search operator. The new algorithm was examined and compared with two well known algorithms on two test functions of different kinds. The results show that the proposed algorithm can find a widespread Pareto front regardless of the shape of the Pareto front.

5. ACKNOWLEDGMENT

The work was supported in part by the Fundamental Research Funds for the Central Universities (No.72005502), China and by the URF (101154/2880) at Victoria University of Wellington.

6. REFERENCES

[1] K. Deb, U. Rao, and S. Karthik. Dynamic multi-objective optimization and decision-making using modified nsga-ii: A case study on hydro-thermal power scheduling. In *Proceedings of Evolutionary Multi-Criterion Optimization*, pages 803–817, 2006.

[2] M. Greeff and A. P. Engelbrecht. Dynamic multi-objective optimization using pso. *In multi-objective swarm intelligent systems, studies in computational intelligence*, 261:105–123, Sept 2010.

[3] Z. Zhang and S. Qian. Artificial immune system in dynamic environments solving time-varying non-linear constrained multi-objective problems. *Soft Computing*, on line, Dec. 2010.