Controller Design of Conflict Multi-objective control problem by Preference

Lihong Xu, Member, IEEE, Bingkun Zhu

Department of Control Science and Engineering, Tongji University,

Shanghai, China, (e-mail: xulhk@163.com, zhu1981_2001@yahoo.com.cn)

Abstract: According to the character that the optimal point is not single in the conflict Multi-Objective Control Problem (MOCP) and optimal solutions cannot be simultaneously obtained by traditional optimization methods in a single simulation run, a new algorithm based on evolutionary computation is presented, which incorporates user's preference information into optimal process for obtaining dense Pareto solutions in preference region and defines a new selection function making control objectives stabilized in this region.

Keywords: Multi-objective control; NSGA-II; User's preference

1. INTRODUCTION

Most real-world control problems naturally involve multiple objectives, and these objectives conflict with each other, so the optimal points cannot be attained simultaneously. In the paste decades and beyond, there are two main methods to solve multi-objective control problem: weighted method and ε-constraint method. In [3]-[6], weighted method is transforming multi-objective into one comprehensive objective. Such as in the optimal control, error, control consumption and destination error these three objectives are weighed to a comprehensive objective. Although this method reduces the problem complexity, obtaining controller cannot reflect essential requirements of these three objectives. The controller would be strongly associated with the trade-off chosen weight; furthermore, weighted method has a fatal shortcoming that it cannot solve non-convex objective space. In [7]-[10], ε-constraint method is presented: choosing one of objective functions as main objective function and the others as the constraints to optimize. Although this method can deal with non-convex objective space, it produces another problem: how to choose appropriate constraint values. Multi-objective optimization is different from single objective optimization. Single objective optimization is not equivalent to multi-objective concordant optimization.

Due to the shortcoming of traditional optimization methods mentioned above, multi-objective control problem also is not equivalent to single objective control problem which can not ensure the existence of a feasible controller in advance, so compatible control framework was presented in [1] and [2]. When these objectives conflict with each other, it is impractical to fix all the objectives at some given optimal points. To ensure the existence of a feasible controller, desired optimal values of all these controlled objectives are backed off to some suboptimal 'interval' or 'region'. For example in the greenhouse control problem, the temperature and humidity can be kept in a certain range that is suitable for plants to grow rather than precise temperature and humidity points. Generally, this certain region is called 'compatible objective region', namely control index of all controlled objectives don't conflict with each other. Multi-Objective Compatible Control (MOCC) hunts for suboptimal and feasible point as the control aim rather than precise optimal point.

To control problem, because different users have different preferences, control problem is different from optimization problem. Then desired control point is different to different user, furthermore user and optimization process should have an interactive relation in control process. User's preference information should direct optimization, whereas results of optimization should feed back to user in order to revise the optimization direction for the next optimization computation. However, this problem is neglected in the traditional optimization algorithm. Besides, how to select an appropriate Pareto solution is a key point in MOCC. In [2], Pareto solution is selected by constraint method. From the objective functions, we choose an objective function as the main objective function and the others are restricted in an interval to optimize. Although this method solves multi-objective control problem in some degree, it can't reflect multi-objective overall essential performance.

Taken these existed problems into account, Conflict Multi-Objective Compatible Control Algorithm Based on Preference (MOCCAP) is presented. In the recent years, evolutionary algorithm develops rapidly and has excellent ability in multi-objective optimization problem. NSGA-II [11] is one of much better algorithms in evolutionary computation. MOCCAP incorporates user's preference information into NSGA-II optimization process in order to obtain a dense Pareto solutions distributed in user's preference region which would help user conveniently choose. Because how to choose a Pareto solution has a significant influence on control performance. So a new selection function is defined－a preference selection function soundly compromising preference and Pareto-optimal.

2. THE PRELIMINARY KNOWLEDGE

Since traditional optimization methods use point-by-point approach which only one optimization direction is modified to the next iteration, it is very difficult for traditional optimization methods to obtain a serial of Pareto optimal solutions in a single simulation run. In recent years, scholars are inspired by the nature's law and find a serial of methods to multi-objective optimization problem. Evolutionary computation is one of these methods, which mimics Darwin's species evolutionary theory to drive its search towards a serial of Pareto optimal solutions. Since 1990, the multi-objective genetic algorithms (MOGA) based on evolutionary computation have received intense attention and been developed rapidly [8], such as TDGA[9], PAES[10], and NSGA-II[11].

The aim of compatible multi-objective optimization is to find a feasible and suboptimal region in objective space. To achieve this, firstly we would introduce some basic knowledge of NSGA-II.

Definition 1: Crowding Distance ([11])

Supposed a,b,c,d are four feasible solutions, which are not dominated with each other. Next, Let two dimensions optimal problem as example:

(1) Compute minimum, $h_1^{\min}$, $h_2^{\min}$ and maximum $h_1^{\max}$, $h_2^{\max}$ of objective function h_1 and h_2 respectively, and assign initial distance $d=0$ of these four solutions;

(2) Sort these solutions a,b,c,d in worse order by value of objective function h_1, supposing the order is a,b,c,d. Because a, d are on the boundary , so set the distances of a, d are ∞; the distances of b, c are:

$$d_b = 0 + \frac{h_1^{(c)} - h_1^{(a)}}{h_1^{\max} - h_1^{\min}};$$

$$d_c = 0 + \frac{h_1^{(d)} - h_1^{(b)}}{h_1^{\max} - h_1^{\min}}$$

(3) Sort these solutions ordered by objective function h_1 in worse order of objective function h_2 again, because of these solutions having the same rank, so the order should be d,c,b,a. Because d, a are on the boundary, so their distances are ∞, the distances of c, b are:

$$d_c = d_c + \frac{h_2^{(b)} - h_2^{(d)}}{h_2^{\max} - h_2^{\min}};$$

$$d_b = 0 + \frac{h_2^{(a)} - h_2^{(c)}}{h_2^{\max} - h_2^{\min}};$$

Crowding distance in NSGA-II not only can keep individuals diversity, but also can prevent 'superman phenomenon' which lead to the algorithm's premature convergence.

Definition 2: Crowded Tournament Selection Operator ([11])

A solution i wins a tournament with another solution j if any of the following conditions are true:

(1) If solution i has a better rank, that is $r_i < r_j$.

(2)If they have the same rank but solution i has a better crowding distance than solution j, that is, $r_i = r_j$ and $d_i > d_j$.

Crowded tournament selection operator can soundly distinguish solutions in multi-objective optimization problems. Indeed, crowded tournament selection operator is a kind of niche skill, moreover which need not bring other new parameters.

3. THE DESCRIPTION OF CONFLICT MULTI-OBJECTIVE COMPATIBLE CONTROL ALGORITHM BASED ON PREFERENCE

3.1 Membership Function and Preference Selection function

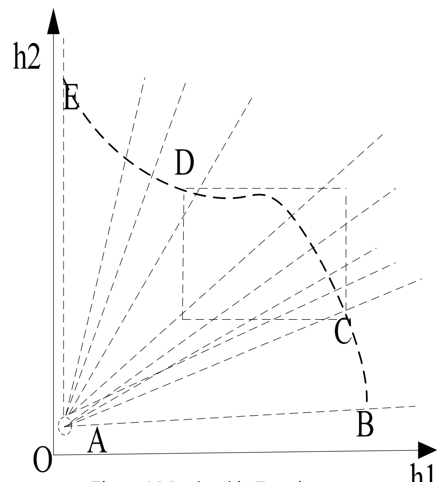

Figure.1 Membership Function

In Figure 1, point A is called 'ideal objective point' [11]. Generally, the ideal objective point corresponds to a non-existent solution. The reason is that the minimal solution of conflict multi-objective optimization for each objective function can not be the same solution. Only when the minimal solution of each objective function is identical, the ideal objective point corresponds to a feasible solution. In this way, multi-objective doesn't conflict with each other. In Figure 1, h_1 and h_2 are objective function; the rectangle is user's preference region; dashed front BE is the Pareto front; the dashed front has two intersection points with preference region in C and D, respectively.

In objective space, Point A as the acme forms many angles which equally divide the preference region into m_1 and other region are equally divided into m_2. In these angles, we define a membership function $f(x)$. Solutions in preference region are defined a higher membership function value than those in other region. Membership function values of solutions in central angle of preference rectangle are defined 1, and from this center angle membership function values of solutions are decreased symmetrically to 0.

Now, a preference selection function $F(x)$ is defined which weighs its membership function value and $1/rank$.

$$F(x) = \omega_1 * f(x) + (1-\omega_1) * 1/rank$$

Obviously, individual in GA has lower rank is better. So here, $F(x)$ weighs its membership function value and $1/rank$ in the preference selection function. A higher value of $F(x)$, a more chance solution x is chosen. In MOCCAP, solutions are selected according to its rank and preference selection function value. In this way, preference selection function can soundly compromise the user's preference and Pareto optimal.

3.2 The Framework of Compatible Control

Conflict multi-objective compatible control can be abstracted into the following theory problem:
Control model:

$$x(k+1) = f(x(k), u(k), k) \tag{1a}$$

$$y(k+1) = g(x(k), u(k), k) \tag{1b}$$

$x(k)$ is system state， $u(k)$ and $y(k)$ are system control input and system control output, k is the control step respectively. All are subject to

$$u(k) \in U \subset \Re^{n_u} \tag{2a}$$

$$x(k) \in X \subset \Re^{n_x} \tag{2b}$$

$$y(k) \in Y \subset \Re^{n_y} \tag{2c}$$

There are n control objectives $h_i(x(k), u(k), k)$, namely

$$\min_{u(k)} (h_1(x(k), u(k), k), ..., h_n(x(k), u(k), k))$$
（3）

$s.t.$ (1), (2)

Compatible control algorithm is described in Fig.2. In MOCCAP, we set the control input $u(k)$ as the optimization variable. Utilizing multi-objective optimization algorithm optimizes $u(k)$, and in the same obtains matched system state $x(k)$ and optimal control law $u(k)$. Next, we let $x(k)$ and $u(k)$ as the initial points go to the next iteration computation. This is different from the ordinary dynamic optimization. The main procedure of MOCCAP is that compatible control algorithm utilizes multi-objective optimal algorithm to optimize and obtains a much better point by preference selection function, and then let the chosen point as the initial point go to the next control step and in the same update the system state. So selection function has a significant influence on the control performance. In order to accelerate algorithm's convergence, the iterated GA is utilized.

3.3 Algorithm:

Based on NSGA-II and preference selection function defined above, MOCCAP algorithm is presented. Because user should have an overall comprehension on the objective space in advance, MOCCAP has two lay computations.

(1) Off-line computation: objective space is obtained by NSGA-II and then let user choose a preference region as control objective region;

(2) On-line computation:

Step1 Initialize parameters: individual number of population $NIND$,max generation $MAXGEN$,max control step $MAXSTEP$, $x(0)$ as the initial state, control input u as optimization variable and create initial population $Chrom$, set gen =0, $k=1$;

Step2 Compute objective function values in initial population, according to function values and domination relation, obtain rank and crowding distance of individual i ;

Step3 Set $Chromparent = Chrom$;

Step4 Make tournament selection operator in $Chrom$, namely two individuals are selected randomly and the better is saved in $Chrom$ according to crowed tournament selection operator;
Step5 Make crossover and mutation operator in Chrom, obtain offspring population $Chromoffspring$;
Step6 Combine parent with offspring populations $Intermediate\_Chrom = Chromparent \bigcup Chromoffspring$ perform non-dominated sorting and compute crowding distance, obtaining $NIND$ Pareto optimal individual from $Intermediate\_Chrom$ composes $Chrom$;
Step7 $gen = gen + 1$, if $gen \le MAXGEN$, return to step3, otherwise return to step 8;

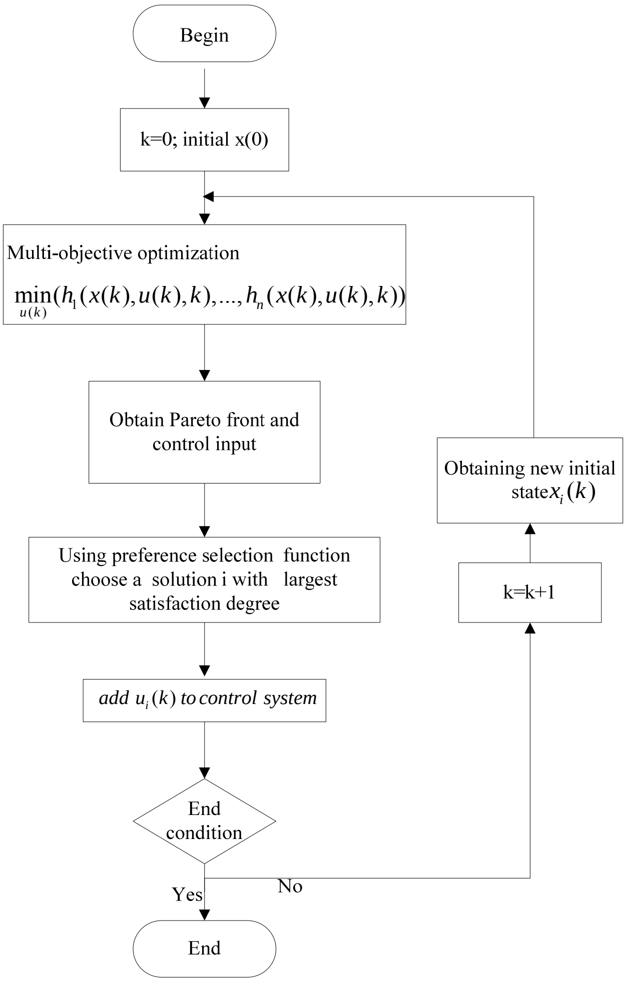

Fig.2 Compatible control flow chart

Step8 Utilize preference selection function select a solution $x(i)$ with largest satisfactory degree and its matched control input u_i;

Step9 Revise the system state error, compare practical control state $x_0(k)$ with predictive objective state $x_p(k)$, set error $\Delta x_0(k) = x_p(k) - x_0(k)$, and then revise the control initial system state $x_0(k+1) = x_0(k) + \Delta x_0(k)$, u_i as control input in the next control step;

Step10 $k = k+1$, if $k < MAXSTEP$, return to step 2, otherwise stop;

4. The Result of Simulation

In order to validate the effectiveness of MOCCAP algorithm, we use an example to illustrate it. To conveniently illustrate our compatible control algorithm, we take a discrete-time system as our controlled model.

System model：

$$x_1(k+1) = -0.2x_1(k)^3 - 0.2x_2(k)^3 + 0.1 - u_1(k)^3 - u_2(k)^3 \quad (4a)$$

$$x_2(k+1) = 0.3x_1(k) - 0.3x_2(k) + 0.1 + u_1(k) - u_2(k) \quad (4b)$$

Objective function：

$$\min \quad h_1(k) = 1 + x_1(k)$$
$$h_2(k) = 1 + x_2(k) \quad (5)$$

Constraints: $u_1, u_2 \in [-0.2, 0.5]$;

Initial points: $x_1(0) = 1, x_2(0) = 1$.

Results of Simulation

First, through off-line computation the preference region is obtained. Here set $h_1(x)$ and $h_2(x)$ is in [0.5， 0.6] and [0.7， 1]respectively. Optimization in compatible control algorithm is different from the ordinary optimization. User's preference information is incorporated into the optimization process. In this way, dense Pareto optimal solutions distributed in the preference region can be obtained. Ordinarily, NSGA-II only can obtain well distributed Pareto solutions. In optimization process, crowding distances of solutions in preference region are magnified in order to achieve requirement of MOCCAP. In Fig.3 and Fig.4, both curves exhibit similar Pareto front. This illustrate optimization in MOCCAP has the same search optimal solutions ability with NSGA-II. Compared Fig.3 with Fig.4, obviously there are dense Pareto optimal solutions distributed in the preference region [0.5， 0.6]× [0.7， 1] in Fig.4.

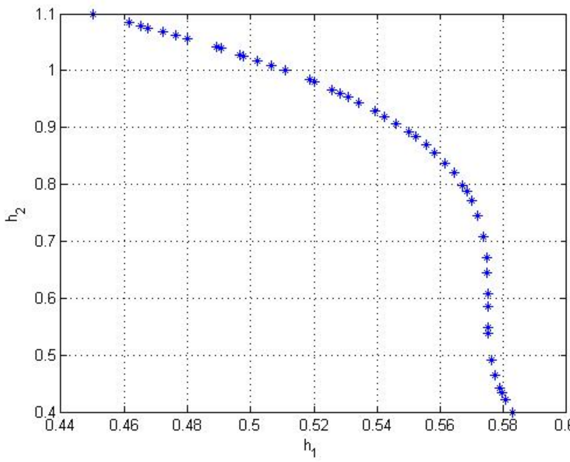

Fig.3 Pareto front of objective space based on NSGA-II

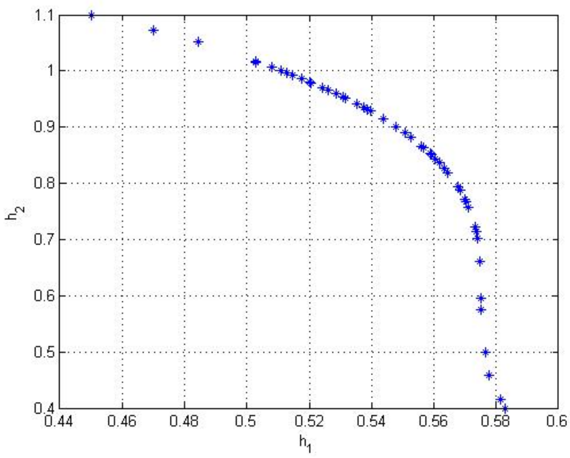

Fig.4 Pareto front of objective space based on preference

In Fig.3 and Fig.4, we can see that objective space of this example is non-convex. To this kind of problem, as we all know, weighted method cannot deal with this kind of question.

ε - constraint method doesn't enable to deal with this kind of problem well. For example, when h_2 is the main objective h_1 as the constraint, we set constraint $h_1 \leq 0.6$. Through optimization, we only can obtain a Pareto solution X^* , $h_2(X^*)$ =0.4. Similarly, other constraint values have the same outcome that only can obtain a Pareto solution. Obviously, it cannot reflect user's preference information and cannot meet user's selection demand.

In multi-objective control problem, there should have much more Pareto solutions for user to choose. So we magnify the crowding distance of solutions of user's preference in computation process. In this simulation, distance of solutions in preference region is four times than solutions in other region. In Fig.4, we obtain dense Pareto solutions distributed in user's preference region. This proves our new algorithm effectiveness and feasibility and meets multi-objective control problem's demand.

In order to illustrate the effectiveness of our new algorithm MOCCAP, we contrast it with MOCC by simulation in Fig.5 and Fig.6. In Fig.5 and Fig.6, horizon axis represents the control step and vertical axis represents control results of objective function.

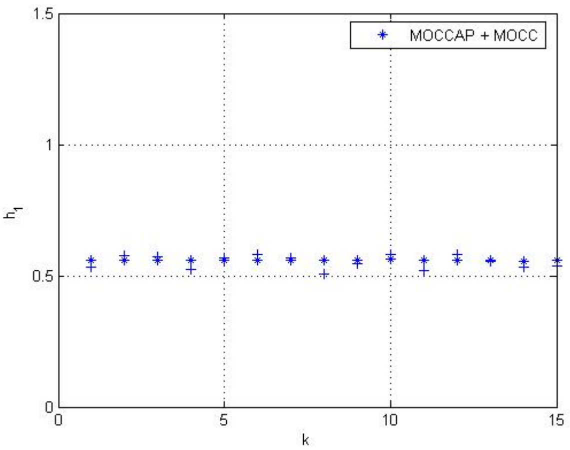

Fig.5 Control result of objective function h_1

In MOCC, objective function h_2 is chosen as the main objective $0.5 \leq h_1 \leq 0.6$ as the constraint condition. In Fig.5 and Fig.6, obviously MOCCAP algorithm not only can make the control objective stabilize, but also the objective function are stabilized in the preference region, respectively; however, the results of MOCC can not meet user's demand. Main objective function h_2 always sways and is not stabilized in the preference region. The reason is that in MOCC algorithm, Pareto solution is selected by constraint method which can not ensure selected objective function values of h_2 always

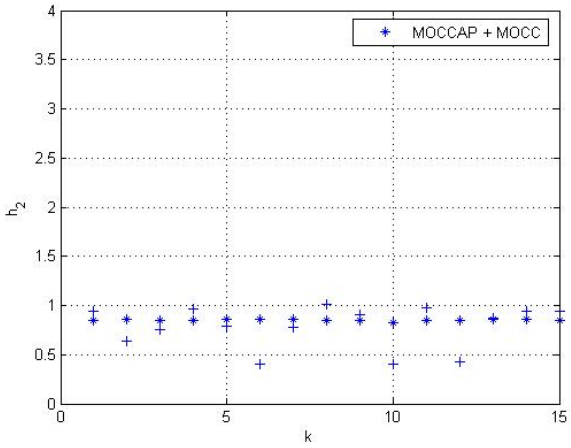

Fig.6 Control result of objective function h_2

in the preference region; but in MOCCAP, Pareto solution is obtained by defined preference selection function, which can ensure that chosen solutions by defined preference selection function with largest satisfaction degree. This illustrates preference selection function has an important role in MOCCAP.

5. CONCLUSION

Through analyzing the questions existed in the conflict multi-objective control problem, this paper brings forward a new multi-objective compatible control algorithm based on user's preference. Firstly obtains preference region by off-line computation with ensuring existence of controller; and then reconstructs NSGA-II which incorporates user's preference information into optimization process and simultaneously utilizes defined preference selection function to select Pareto solution. This algorithm indeed solves multi-objective optimization problem by multi-objective optimization method. So it avoids shortcomings and disadvantage of transforming multi-objective control problem to a comprehensive single objective control problem.

REFERENCES

Lihong Xu, Zhiqiang Zou, and Qingsong Hu, Two-Layer Optimization Compatible Control for Multi-Objective Control Systems, IEEE International Conference on Networking, Sensing and Control, 23-25 April, 2006, pp.658-663.

Lihong Xu, Qingsong Hu and Erik Goodman. Two layer Iterative Multi-objective Compatible Control algorithm. the 46th IEEE Conference on Decision and Control,2007, pp2992-2997.

Masaaki, I. (1997), Multi-objective optimal control through linear programming with interval objective function, SICE, July 29-31, Tokushima, pp. 1185-1188.

Rangan, S. &Poolla, K.(1997), Weighted optimization for multi-objective full-information control problems, System & Control Letter, Vol.31, pp. 207-213.

Eisenhart, K. J. (2003), Multi-objective optimal control problems with endpoint and state constraint, Ph.D. thesis, Western Michigan University.

Iskander M. G.(2003), Using different dominance criteria in stochastic fuzzy linear multi-objective programming: A case of fuzzy weighted objective function, Mathematical and Computer Modeling, Vol. 37,pp. 67-176.

Zhao, Z., Yasuyuki, I. & Masatoshi, N.(2003), Diffuser multi-objective control for STEC plant, IEEE Conference on Systems, Man and Cybernetics, October 5-8, 2003, pp. 148-153.

Masaaki, I. (1997), Multi-objective optimal control through linear programming with interval objective function, SICE, July 29-31, Tokushima, pp. 1185-1188.

Scherer, C. W.(1995), Multi-objective H_2/H_1 Control, IEEE Transaction on Automatic Control, vol. 40, pp. 1054-1061.

Scherer, C, Gahinet P. & Chilali, M.(1997), Multi-objective output feedback control via LMI Optimization, IEEE Transactions on Automatic Control, vol. 42, pp. 896-911.

Kalyanmoy Deb (2002), Multi-Objective Optimization using Evolutionary Algorithms, JOHN WILEY & SONS, LTD, pp. 235-237.