WCCI 2012 IEEE World Congress on Computational Intelligence
June, 10-15, 2012 - Brisbane, Australia IEEE CEC

Mapping Constrained Optimization Problems to Algorithms and Constraint Handling Techniques

Chengyong Si, Lei Wang, *Member, IEEE*, Qidi Wu, *Senior Member, IEEE*

***Abstract*-During the past few decades, many Evolutionary Algorithms together with the constraint handling techniques have been developed to solve the constrained optimization problems which have attracted a lot of research interest. But it's still very difficult to decide when and how to use these algorithms and constraint handling techniques effectively. Some researchers have proposed some general frameworks like population-based algorithm portfolios (PAP), cooperative coevolving or ensemble strategies which use different subpopulations to run the algorithm parallel. These ideas don't consider the problems' characteristics in detail. Motivated by these observations, we propose a new method to construct the relationship between problems and algorithms as well as the constraint handling techniques standing the qualitative and quantitative point of view. This paper first summaries and extracts the problems' characteristics systematically, then combines different qualitative and quantitative methods in the Evolutionary Algorithms and constraint handling techniques respectively so as to get a reasonable correspondence. The experimental results confirm this relationship, which is valuable to guide future research.**

Keywords-Constrained optimization; Problem characteristics; Qualitative and quantitative methods; Particle Swarm Optimization

I. INTRUDUCTION

Constrained Optimization Problem, or Nonlinear Programming Problem (NLP), can widely occur in many economic, technical and scientific projects. But solving the constrained optimization problem remains a difficult and open question for Evolutionary Algorithms (EAs) [1].

Although numerous population-based algorithms, such as Evolutionary Computation (EC) [2], Particle Swarm Optimization (PSO) [3], Ant Colony Optimization (ACO) [4], Cultural Algorithms (CA) [5] are developed to solve the constrained optimization problems, there is no best algorithm for all problems. On the other hand, Constrained Optimization Evolutionary Algorithms (COEAs) can be considered as constraint handling techniques plus EAs [6]. Therefore, an effective constraint handling technique needs to be in conjunction with an efficient EA to obtain competitive performance.

Many researchers improve the algorithms and constraint handling techniques just from the running results [7]-[13], without full study of the problem's characteristics which may directly reflect the algorithm's performance.

Recently, Peng *et al.* [14] proposed a new approach named population-based algorithm portfolios (PAP). The basic idea is that it would be less risky to distribute the time among multiple different algorithms. PAP runs each constituent algorithm with a part of the given time. Besides, when choosing the offspring, a different strategy is proposed. Migration is the only route that different subpopulations communicate with each other, which will reduce the likelihood of different constituent algorithms repeating similar search behaviors or sharing similar search biases. Although this approach may get a satisfying result, it is only a compromising result and there are some problems to be solved, for example, how to determine the number of algorithms, how to distribute the time on each algorithm.

Besides, Yao and his colleagues or other students have done a lot of similar work using the concept of cooperative coevolution [15]-[17]. Yang *et al.* [15] proposed a new cooperative coevolution framework which introduces the random grouping scheme and adaptive weighting in population decomposition and coevolution. The main aim is to solve the nonseparable problem for which tight interactions exist among different decision variables when decomposing a high-dimensional problem into single dimensions. Li and Yao [16] present a cooperative coevolving PSO (CCPSO) algorithm. This algorithm outperforms CPSO proposed by van den Bergh and Engelbrecht [18], which has two new cooperative PSO modes, namely CPSO-S_k and CPSO-H_k. Based on CCPSO, Li and Yao "in press" [17] proposed CCPSO2, which adopts a new PSO position update rule that relies on Cauchy and Gaussian distributions and a scheme to dynamically determine the coevolving subcomponent sizes of the variables. It can solve large-scale optimization problems.

As there are several alternative approaches or strategies at every step of an Evolutionary Algorithm and parameters in each approach may require the users to fine tune according to different problems, P.N. Suganthan with his colleagues and students proposed the idea of "ensemble" to benefit from both the availability of diverse approaches and the need to tune the associated parameters [19].

For example, as to the ensemble of different operators and parameters, Mallipeddi *et al.* [20] proposed the ensemble strategies with adaptive evolutionary programming, which mainly benefit from different mutation operators with different parameter values during different strategies. Also in this method, each mutation operator has its own population. More importantly, the strategy parameter values can be adaptively selected according to the search performance in the previous generations. Pan *et al.* [21] proposed a harmony search algorithm with ensemble of parameter sets (EHS). It adopts the best control parameters self-adaptively during the evolution process with the different parameter values combination prepared beforehand.

The authors are with the College of Electronics and Information Engineering, Tongji University, Shanghai 201804, China (e-mail: sichengyong@163.com, wanglei@tongji.edu.cn).

Besides the ensemble of some small strategies during the algorithm, some other kinds of ensemble like constraint handling techniques and algorithms were also proposed.

Mallipeddi and Suganthan [22] proposed an ensemble of constraint handling techniques (ECHT), which utilizes multiple subpopulations. Each subpopulation is corresponding to one constraint handling method, which produces its own offspring and evaluates them. But the parent population will not only compete with its own offspring population but also with offspring population of the other constraint handling methods. So in ECHT, every function call is utilized effectively. Yu and Suganthan [23] presented the idea of ensemble of niching algorithms (ENA) using four different parallel populations. Like ECHT, the offspring of each population is considered by all parallel populations.

He and Yen [24] proposed an ensemble method for performance metrics in multiobjective EAs, which can provide a comprehensive measure for MOEAs.

All of these ideas can get relatively satisfying results. But the problem's characteristics are little considered, that means the results are mostly "compromising". If the researcher doesn't know the relationship between the problem and algorithm together with the constraint handling techniques, these approaches may be suitable. But if the algorithm and the problem can be mapped together, or in other words, if we know which kind of algorithm and constraint handling technique is suitable to solve the problem, then the solution process may become very simple and some ensemble techniques can be added. So it may be a good method to study the problem's characteristics first. That is what we will do in this paper. This work will also be helpful for future research.

Tsang and Kwan [25] pointed out the need to map constraint satisfaction problems to algorithms and heuristics. But they didn't give an exact relationship between them. This paper will try to find the corresponding relationship between problems and algorithms together with constraint handling techniques from qualitative and quantitative point of view. To do this work, some problem pool which contains problems with different characteristics is needed.

The rest of this paper is organized as follows. Section II describes the problem's characteristics in detail. Section III presents the combination of qualitative and quantitative methods used in EAs and constraint handling techniques respectively. The experimental results and analysis are presented in Section IV. Section V gives the concluding remarks.

II. CONSTRAINTED PROBLEM'S CHARACTERISTICS

Depending on whether there are constrained functions, optimization problems can be classified into two categories: unconstrained optimization and constrained optimization. Both categories give the range of each variable, but the constraints on the variables are not given in unconstrained optimization..

In the following definitions, the goal of the problems are assumed (without loss of generality) minimization.

The general NLP can be expressed as follows [6]:

$$\text{Minimize } f(X) \tag{1}$$

Subject to:

$$g_j(X) \leq 0;\ j=1,\dots,l \tag{2}$$

$$h_j(X)=0;\ j=l+1,\dots,p \tag{3}$$

where $X=(x_1,x_2,\dots,x_n)$ is the vector of solution, and each x_i $(i=1,\dots,n)$ is bounded by lower and upper limits $l_i \leq x_i \leq u_i$ which define the search space S; l is the number of inequality constraints and $p-l$ is the number of equality constraints. The constraints define the feasible region F.

For an inequality constraint that satisfies $g_j(X)=0;\ j=1,\dots,l$ at any feasible point X, it can be called *active* at X. All equality constraints $h_j(X)=0;\ j=l+1,\dots,p$ are considered *active* at all points of F.

When solving NLPs with EAs, equality constraints are usually converted into inequality constraints of the form:

$$g_j(X)=|h_j(x)|-\delta \leq 0;\ j=l+1\dots,p \tag{4}$$

where δ is a tolerant value(often a very small positive value).

By now, no effective theoretical results on the characteristics that make a constrained problem difficult for an evolutionary algorithm (or other methods) are available. As to the study of constrained optimization problems, many researchers improve the algorithms just from the running results, without full study of the problem's characteristics. This paper attempts to summary the problem's characteristics, and compare it to the traditional cultural model so as to take different strategies to optimize the problems which can reduce the risk associated with the selection of algorithms.

Constrained optimization problems can be characterized by different parameters: the number of linear constraints, the number of nonlinear constraints, the number of equality constraints, the number of local optimal solution, and so forth.

This paper attempts to adopt another standard for describing from the nature of the variables in the constraints. The four elements are as follows:

A. the number of variables

The number of variables will directly determine the possible computational complexity. For the limited operation rules, the number of variables is a basic condition to judge an optimization problem complex or not. An optimization problem with more variables is more likely to be complex than one with fewer variables.

B. variable index

Variable index refers to the variable's own index, such as x_1^2, x_3^5 and so on. This is to identify the problem's possible complexity without considering each variable's interaction with other variables. In general, the greater the index, the more complex a problem may become.

C. the interrelationship between variables

The interrelationship between variables refers to the interactive computing among variables, such as x_1x_3, $x_1^3(x_1+x_2)$ and so on. Here, the interdependence of different variables is mainly concerned. It's the core reflection of the complexity of the constrained optimization problems, because the nature of constrained problems is to limit the satisfying constraints among variables. From that, it can be inferred that the interrelationship is more important in the constraints than in the objective functions.

D. the impact of the constant

Here, the constants mainly refer to the ones that are directly together with the variables, causing a variable amount of migration, such as -0.25 in the expression of $\sin(-x_3-0.25)$ and so on. Besides, some separate constant in the constraints are also included, such as 1 in the expression of $x_1^2 - x_2 + 1 \le 0$, but its impact is less than the previous one. These constants will have an enormous impact on the results, because although they are the direct embodiment of the offset, they can not always achieve consistency and balance between the objective function and constraints.

III. DESCRIPTION OF QUALITATIVE AND QUANTITATIVE METHODS

Qualitative and quantitative methods here can be regarded as inaccurate and precise methods used in the EAs and constraint handling techniques respectively, which will have different influences on the problems to be solved.

The basic idea is that through different combinations of qualitative and quantitative methods to solve different problems, the problem types that they are good at solving can be got as to the EAs and constraint handling techniques, and then a corresponding relationship can be constructed.

A. Qualitative and Quantitative Methods in Algorithms

The basic algorithms are based on Cultural Algorithm based PSO (CAPSO)[26].

Inertia weight *w* is an important parameter which can provide a proper balancing mechanism to adjust the balance between exploration and exploitation. When *w* is larger, it has a strong ability for exploration, suitable for searching for a wide range at the beginning of the search, so called global search; when *w* is smaller, it has a better ability to exploit, suitable for more refined search at the end of the search, so called local search [27]. Currently, the methods of adjusting inertia weight include: linear adjustment, non-linear adjustment and random adjustment.

The parameter of inertia weight can have the qualitative and quantitative forms. In this paper, the qualitative form adopts the step function while the quantitative form adopts the non-linear decreasing.

The inertia weight *w* in both forms is larger at the beginning and smaller at the end which changes over time. For easily expressing, the parameter *t* is introduced, which corresponds to the generation, and it's set to (0,1). The equation and relationship between *t* and generations can be seen from (5) and Fig.1 .

$$t = \frac{i_{ter}}{i_{ter\_\max}} \tag{5}$$

where i_{ter} and i_{ter_max} represent the generation and maximum generation respectively.

The qualitative form of inertia weight (*w1(t)*) is set to change according to (6) while the quantitative form (*w2(t)*)is set to change according to (7).

$$w1(t) = \begin{cases} 0.90 & 0.0 \le t < 0.2 \\ 0.72 & 0.2 \le t < 0.4 \\ 0.54 & 0.4 \le t < 0.6 \\ 0.36 & 0.6 \le t < 0.8 \\ 0.18 & 0.8 \le t < 1.0 \end{cases} \tag{6}$$

$$w2(t) = (1 - t^2) \times (w_{\max} - w_{\min}) + w_{\min} \tag{7}$$

where w_{min} and w_{max} represent the minimum and maximum values of the inertia weight respectively. In this experiment, w_{min} is set to 0 while w_{max} is set to 0.9.

The relationship between *w1(t)* and *w2(t)* are shown in Fig. 2 and Fig. 3 respectively.

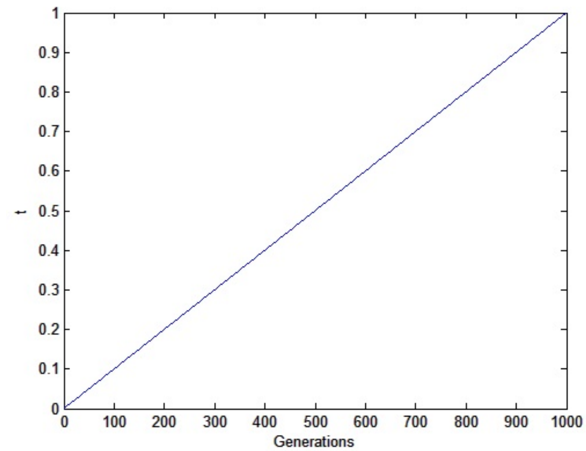

Fig. 1. The relationship between *t* and i_{ter} (Generations)

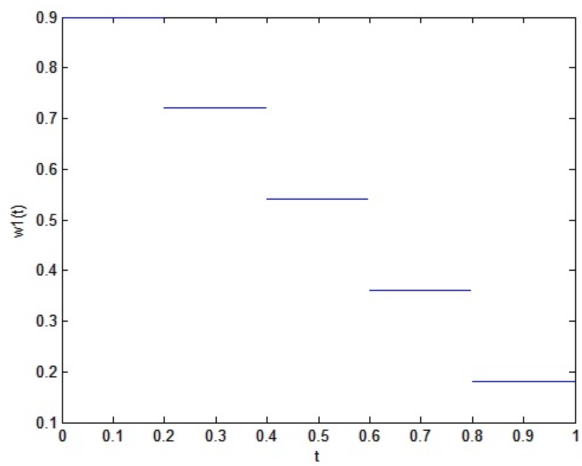

Fig. 2. The relationship between t and w1(t)

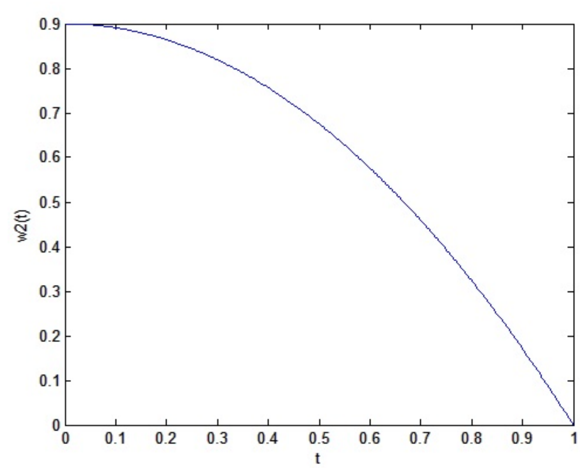

Fig. 3. The relationship between t and w2(t)

B. Qualitative and Quantitative Methods in Constraint Handling Techniques

The qualitative and quantitative methods in the constraint handling techniques are mainly reflected in the evaluation function. For qualitative method, the evaluation function uses the objective function together with the feasibility of the solutions. For quantitative method, the evaluation function is based on simulated annealing penalty function [28], as shown in (8).

For qualitative method, if the solution is feasible, then the evaluation function value is the objective function value, or the evaluation function value will be different large values (the problems are assumed minimization) according to the number of violation constraints, which is beneficial to sorting and selecting.

For quantitative method, the simulated annealing penalty method is adopted. All individuals are unified. The evaluation function is not designed for feasible and infeasible solutions respectively but using the violation of penalty function for all individuals.

The equation for evaluation function is as follows:

$$\text{eval}(x) = f(x) + \frac{1}{2\tau}\phi(x) = f(x) + \frac{1}{2\tau}\sum_{j=1}^{p} f_j^2(x) \tag{8}$$

where τ is the temperature(later discussed in detail). $f_j(x)$ describes the violation of the *j-th* constraint, the equation is as follows:

$$f_j(x) = \begin{cases} \max\{0, g_j(x)\},\ 1 \le j \le l \\ \max\{0, |h(x)| - \delta\},\ l+1 \le j \le p \end{cases} \tag{9}$$

The design of τ is important when using the simulated annealing penalty method. As our main aim is to check the performance of different combinations of qualitative and quantitative methods, we just adopt to a simple way in which τ is set to constant.

C. Different Combinations

To better understand the influence of qualitative and quantitative methods in algorithms and constraint handling techniques respectively, we design 4 combinations, including the qualitative algorithm with the qualitative and quantitative constraint handling techniques (A_{ql}-C_{ql} and A_{ql}-C_{qn}) and the quantitative algorithm with the qualitative and quantitative constraint handling technique (A_{qn}-C_{ql} and A_{qn}-C_{qn}).

Through the comparison between different algorithms with the same constraint handling technique (A_{ql}-C_{ql} and A_{qn}-C_{ql}, A_{ql}-C_{qn} and A_{qn}-C_{qn}), the different problem characteristics for A_{ql} and A_{qn} can be got. Similarly, the corresponding problem characteristics for C_{ql} and C_{qn} can be obtained.

IV. EXPERIMENTAL STUDY

In this section, different combinations presented in Section III are empirically evaluated on 13 well-known benchmark test functions used in [29].

A. Benchmark Functions

The benchmark functions chosen contain characteristics that are representative of what can be considered "difficult" global optimization problems for an evolutionary algorithm. The characteristics of these benchmark functions are summarized in Table I, where the number of variables n, the type of the function f, the relative size of the feasible region in the search space given by the ratio ρ, the number of constraints of each category (linear inequalities LI, nonlinear equations NE and inequalities NI) and the number a of active constraints at the optimum (including equality constraints) are listed.

To get an estimate of how difficult it is to generate the feasible points through a random process, the metric ρ (as suggested by Michalewicz and Fogel [1]) is computed using the following expression:

$$\rho = \frac{|F|}{|S|} \tag{10}$$

where |S| is the number of randomly generated solutions and |F| is the number of feasible solutions found.

The values of ρ are also listed in Table I [30].

TABLE I
SUMMARY OF THE BENCHMARK FUNCTIONS

f	**n**	**Type of f**	**ρ(%)**	**LI**	**NI**	**LE**	**NE**	**a**
g01	13	quadratic	0.0111	9	0	0	0	6
g02	20	nonlinear	99.9971	0	2	0	0	1
g03	10	polynomial	0.0000	0	0	0	1	1
g04	5	quadratic	51.1230	0	6	0	0	2
g05	4	cubic	0.0000	2	0	0	3	3
g06	2	cubic	0.0066	0	2	0	0	2
g07	10	quadratic	0.0003	3	5	0	0	6
g08	2	nonlinear	0.8560	0	2	0	0	0
g09	7	polynomial	0.5121	0	4	0	0	2
g10	8	linear	0.0010	3	3	0	0	6
g11	2	quadratic	0.0000	0	0	0	1	1
g12	3	quadratic	4.7713	0	1	0	0	0
g13	5	nonlinear	0.0000	0	0	0	3	3

B. Experimental Settings

All the algorithms are run 30 times on each test problem with maximum number of function evaluations (Max_FEs) in each run set to 50 000.

To maintain consistency, the parameters in the basic algorithm and constraint handling techniques are the same. The parameters used in the CAPSO are set as follows: w_{min} = 0, w_{max} =0.9, c_1=c_2=2, population size =50, acceptance rate p=0.3, update frequency k=5 which means the belief space will be updated every 5 generations. As to the offspring selection, tournament selection method is used between the parent populations and offspring populations.

TABLE II
COMPARISON OF QUALITATIVE AND QUANTITATIVE METHODS COMBINATIONS

Func. & Optimal Value		A_{ql}-C_{ql}	A_{qn}-C_{ql}	A_{ql}-C_{qn}	A_{qn}-C_{qn}
G01 -15.0000	best	-15.0000	-14.3957	-15.0000	-15.0000
	median	-15.0000	-11.9140	-15.0000	-12.5000
	mean	-14.5818	-11.1890	-14.6302	-12.6080
	st.dev.	1.2E+00	2.1E+00	8.4E-01	2.3E+00
	worst	-9.4531	-6.0000	-12.4531	-9.0000
	FR	100%	100%	0%	0%
G02 -0.803619	best	-0.772978	-0.754749	-0.785132	-0.767316
	median	-0.604612	-0.518724	-0.569570	-0.562133
	mean	-0.598089	-0.527116	-0.589652	-0.565140
	st.dev.	1.1E-01	9.3E-02	1.2E-01	1.1E-01
	worst	-0.359576	-0.402550	-0.428203	-0.387458
	FR	100%	100%	47%	100%
G03 -1.0005	best	-0.9957	-0.9980	-0.0988	-0.7592
	median	-2.0E-04	0	-0.0039	-1.4893
	mean	-0.2844	-0.1209	-0.0216	-0.0721
	st.dev.	4.9E-01	3.3E-01	3.3E-02	1.6E-01
	worst	-2.8E-07	0	-1.3E-07	0
	FR	23%	53%	50%	93%
G04 -30665.5387	best	-30665.5387	-30665.5387	-30665.6204	-30665.6270
	median	-30665.5387	-30661.5325	-30665.6204	-30665.6204
	mean	-30665.5387	-30617.6671	-30665.6204	-30665.5434
	st.dev.	1.0E-11	9.5E+01	2.6E-07	3.2E-01
	worst	-30665.5387	-30185.4777	-30665.6204	-30663.9573
	FR	100%	100%	0%	0%
G05 5126.4967	best	5127.0570	5444.5384	5202.3468	5480.3032
	median	5828.2054	7372.7169	5900.3262	5800.3907
	mean	6007.6250	7079.6017	5774.7334	5854.8763
	st.dev.	7.3E+02	1.2E+03	3.4E+02	2.3E-01
	worst	7051.7663	8128.4347	6112.0169	6112.0917
	FR	0%	0%	70%	33%
G06 -6961.8139	best	-6961.8139	-6961.8139	-6962.0854	-6962.0854
	median	-6961.8139	-6961.8139	-6962.0854	-6962.0854
	mean	-6961.8112	-6812.8113	-6962.0854	-6962.0854
	st.dev.	1.4E-02	7.8E+02	1.0E-06	1.8E-06
	worst	-6961.7404	-2794.1603	-6962.0854	-6962.0854
	FR	90%	93%	0%	0%
G07 24.3062	best	24.3070	24.4010	24.3502	24.7012
	median	24.6634	25.6241	25.0129	26.1249
	mean	24.7997	65.3674	25.2552	45.9900
	st.dev.	4.1E-01	7.9E+01	1.0E+00	4.6E+01
	worst	25.9074	329.1791	26.9189	183.6542
	FR	50%	70%	30%	60%
G08 -0.09582504	best	-0.09582504	-0.09582504	-0.09582504	-0.09582504
	median	-0.09582504	-0.09582504	-0.09582504	-0.09582504
	mean	-0.09582504	-0.09359487	-0.09582504	-0.08470417
	st.dev.	1.7E-17	1.2E-02	1.7E-17	2.5E-02
	worst	-0.09582504	-0.02914380	-0.09582504	-0.02914380
	FR	100%	100%	100%	100%

TABLE II
(CONTINUED)

G09 680.630057	best	680.640407	680.635116	680.637498	680.644391
	median	680.693056	681.139417	680.670450	680.702971
	mean	680.716227	720.842998	680.678762	680.866355
	st.dev.	7.5E-02	6.8E+01	3.2E-02	5.5E-01
	worst	680.921494	942.243697	680.7621	683.083719
	FR	100%	100%	60%	67%
G10 7049.2480	best	7469.0462	7378.3325	7142.3230	7189.4679
	median	7759.4005	8535.0588	7375.2371	8529.7793
	mean	7995.1229	8562.6386	7363.5707	9363.6556
	st.dev.	6.8E+02	8.6E+02	1.1E+02	3.0E+03
	worst	9082.5967	9878.0879	7615.1056	21000
	FR	27%	27%	0%	0%
G11 0.7499	best	0.7499	0.7499	0.7499	0.7499
	median	0.7499	0.7499	0.7499	0.7499
	mean	0.7644	0.7627	0.7499	0.7500
	st.dev.	5.5E-02	5.1E-02	9.4E-06	7.7E-04
	worst	0.9792	0.9999	0.7499	0.7541
	FR	100%	100%	30%	0%
G12 -1.0000	best	-1.0000	-1.0000	-1.0000	-1.0000
	median	-0.9972	-1.0000	-1.0000	-1.0000
	mean	-0.9865	-0.9996	-0.9992	-1.0000
	st.dev.	2.7E-02	1.4E-03	3.2E-03	1.6E-06
	worst	-0.8781	-0.9944	-0.9864	-1.0000
	FR	100%	100%	60%	83%
G13 0.0539415	best	0.0779215	0.0748900	0.0732536	0.4170143
	median	1.1631854	0.9999600	0.9300116	0.9200808
	mean	2.2915550	0.7868909	1.3192481	1.0531518
	st.dev.	2.4E+00	4.0E-01	2.3E+00	8.0E-01
	worst	9.2035677	1.5788700	11.8553302	4.3392085
	FR	0%	0%	77%	73%

As we mainly concerned about the performance of different types of algorithms and constraint handling techniques, we made strict handling for constraint violations. The tolerant value is never used in inequality constraints. In equality constraints, the tolerant value is set to 10^{-4} when converting them into inequality constraints. This strategy will cause the number reduction of feasible values and some feasibility rate (FR) in certain problems may even be 0%. But this is necessary for the performance analysis. Besides, as to the qualitative method for constraint handling techniques, the performance is judged through the number of violation constraints m, described as m*A (A is a large value), which means if both individuals are infeasible ones, the one with less number of violation constraints is better.

C. Experimental Results and Discussions

Table II presents the results obtained by the different combinations of qualitative and quantitative methods in algorithms and constraint handling techniques in 30 independent runs, including the best, median, mean, standard deviation and worst values for each test function.

It should be pointed out that to highlight the performance of different combinations of qualitative and quantitative methods in algorithms and constraint handling techniques, only basic operations are adopted, which may cause the results not so satisfying.

The results here are all based on feasible solutions. As to problems using the quantitative methods, many of the results are infeasible (with FR=0%), but the violation is so small and as the performance comparison happens in the same problem, the infeasible value statistics results are also presented here.

To fully compare the performance of different methods in algorithms and constraint handling techniques, the results will be analyzed in three aspects: 1) the comparison of different algorithms using the same constraint handling technique; 2) the comparison of different constraint handling techniques using the same algorithm.

1) *Comparison of Different Algorithms Using the Same Constraint Handling Technique:* All six combination methods could find optimal values in problems G08, G11 and G12 and nearly optimal values in problems G01, G02, G04, G07 and G09.

FR with the same constraint handling technique is nearly the same, which is good for the algorithm comparison.

As to problems G01, G04, G05, G06, G07, G08, G09, G10 and G11, the qualitative method appear to have a better performance in all evaluation indexes except that the FR with quantitative method in G7 is larger than that with qualitative method. As to problems G02, G03 and G12, the FR with quantitative method is larger and the quantitative method appears to have a better standard deviation. As to problem G13, the quantitative method appears to have a better median value, mean value, worst value and standard deviation.

In conclusion, the better performance of qualitative method in algorithms could be observed in G01, G04, G05, G06, G07, G08, G09, G10 and G11 while the quantitative method performs better in G02, G03, G12 and G13.

2) *Comparison of Different Constraint Handling Techniques Using the Same Algorithms:* The results of different constraint handling techniques using the same algorithms are quite different from the above comparison.

The FR varies greatly among different constraint handling methods in problems G01, G04, G05, G06, G10, G11 and G13 and nearly the same in problems G02, G07 and G08.

As to G01, G04, G06 and G10, FR with qualitative method is much larger than that with quantitative method which is around 0. The objective values in quantitative methods are better than that in qualitative method or even better than the known optimal values but the cost is constraint violation.

As to G05 and G13, the quantitative method could obtain better results as to the median values, mean values, standard deviation and a much better FR. The same thing of FR and standard deviation happens to G03 with worse median and mean values.

As to G09, G11 and G12, the qualitative method could obtain a better FR with worse median values, mean values and standard deviation.

As to G02 and G07, the performance in both methods is nearly the same.

In conclusion, the better performance of qualitative method in constraint handling techniques could be observed in G01, G04, G08 and G10 while the quantitative method performs better in G03, G05, G09, G11, G12 and G13. As to G02, G06 and G07, both of the methods perform nearly the same.

D. Corresponding Relationship

Table III shows the characteristics of the test functions, including variable number, variable index, interrelationship and constant.

In the last three columns, the first number means the statistics in the objective function while the second number means the statistics in the constraint. For example, 3/1(sin) means the max variable index is 3 in the objective function and 1 in the constraints with sine functions existed. In the last two columns, if there are constants and interrelationships between variables in the objective functions or constraints, then the number will be 1, or else 0. We say the objective function has more complex characteristics in interrelationship and constant if the number value of them in objective function is larger than that in constraints.

TABLE III
CHARACTERISTICS OF THE TEST FUNCTIONS

Problem	Variable number	Variable index	Interrelationship	Constant
g01	13	2/1	0/0	0/0
g02	20	4(cos)/1	1/1	0/0
g03	10	1/2	1/0	0/0
g04	5	2/2	1/1	0/0
g05	4	3/1(sin)	0/1	0/1
g06	2	3/2	0/0	1/1
g07	10	2/2	1/1	1/1
g08	2	3(sin)/2	1/0	0/1
g09	7	6/4	1/1	1/0
g10	8	1/1	0/1	0/0
g11	2	2/2	0/0	1/0
g12	3	2/2	0/0	1/1
g13	5	1(exp)/3	1/1	0/0

To sum up, following the conclusion obtained in Section IV, the qualitative method in algorithms is more suitable to solve problems in which the variable index in objective function is larger than that of constraints and the objective function has simpler or same characteristics in interrelationship and constant as constraints; the quantitative method in algorithms is more suitable to solve problems in which the objective function has more complex characteristics in interrelation and the same characteristics in constant as constraints and the variable index in constraints is larger than or equal to two with the variable number larger than or equal to three.

As to the constraint handling techniques, the qualitative method is better at solving problems which has the same characteristics of constant in the objective function and constraints while the index of objective function is larger than or equal to that of constraints; the quantitative method is better at solving problems which contain the equality constraints or the index of objective function is less than that of constraints while the interrelationship in constraints are more complex than that of objective function.

V. CONCLUSION

This paper has presented a new way to construct the relationship between problems and algorithms together with the constraint handling techniques, which is based on the problems' characteristics.

As different combinations of algorithms and constraint handling techniques has varied performance on a set of problems, which means one combination can't performance well over all problems, we try to get the corresponding problem characteristics for different algorithms and constraint handling techniques from the qualitative and quantitative point of view. Based on this, four different combinations are designed and evaluated on 13 well-known benchmark test functions with different characteristics and the mapping is finally obtained which provides a new way for the constrained optimization research. As this research can be viewed as the first and basic step for constrained optimization, even some small conclusion will be much helpful to the entire optimization.

The future work of this study includes the diversity of the problem pool and more complex qualitative and quantitative methods, or other classifications. And the generalizations about the characteristics of problems should be tested with more problems and theoretically supported.

ACKNOWLEDGMENT

The authors gratefully acknowledge the support of the National Natural Science Foundation of China under Grants 70871091, 61075064, 61034004 and 61005090, the Ph.D. Programs Foundation of Ministry of Education of China under Grant 20100072110038 and the Program for New Century Excellent Talents in University of Ministry of Education of China for this research.

REFERENCES

[1] Michalewicz Z. and Fogel D. B., *How to solve it: modern heuristics*. Berlin, Germany: Springer-Verlag, 2004.

[2] J. Kennedy, R.C. Eberhart and SHI Yuhui, *Swarm intelligence*. Beijing, China: Posts & Telecom Press, 2009.

[3] R.C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proc. 6th Int. Symp. Micromachine Human Sci.(MHS)*, Piscataway, NJ, 1995, pp. 39-43.

[4] A. Colorni, M. Dorigo and V. Maniezzo, "Distributed optimization by ant colonies," in *Proc. 1st Eur. Conf. Artif. Life*, Paris, France, 1991, pp. 134-142.

[5] R.G. Robert, "An introduction to Cultural Algorithms," in *Proc. 3rd Ann. Conf. Evol. Prog.*, RiverEdge, NJ, 1994, pp. 131-136.

[6] Y. Wang, Z. Cai, Y. Zhou and Z. Fan, "Constrained optimization based on hybrid evolutionary algorithms and adaptive constraint-handling technique,"*Struct. Multidisc. Optim.*, Vol. 37. no. 4, pp. 395-413, Apr. 2009.

[7] X. Liang, W. Long, H. Qin, S. Li and G. Yan, " Constrained optimization evolutionary algorithm based on individual feasibility of population," *Control and D.*, Vol. 25. no. 8, pp. 1129-1132,1138, Aug. 2010.

[8] T. P. Runarsson and X. Yao, "Search biases in constrained evolutionary optimization," *IEEE Trans. Syst. Man. Cybern.*, Vol. 35. no. 2, pp. 233-243, May 2005.

[9] A. H. Aguirre, S. B. Rionda and C. A. C. Coello, "Handling constraints using multi-objective optimization concepts," *Int. J. Numer. Meth. Eng.*, Vol. 59. no. 15, pp. 1989-2017, Apr. 2004.

[10] Y. Wang, Z. Cai, G. Guo and Y. Zhou, "Multiobjective optimization and hybrid evolutionary algorithm to solve constrained optimization problems," *IEEE Trans. Syst. Man. Cybern.*, Vol. 37. no. 3, pp. 560-575, Jun. 2007.

[11] L.Guillermo and C. A. C. Coello, "Boundary search for constrained numerical optimization problems with an algorithm inspired by the ant colony metaphor," *IEEE Tran. Evol. Comput.*, Vol. 13, no. 2, pp. 350-368, Apr. 2009.

[12] Y. G. Woldesenbet, G. G.Yen and B. G. Tessema, "Constraint handling in multiobjective evolutionay optimization," *IEEE Tran. Evol. Comput.*, Vol. 13, no. 3, pp. 514-525, Jun. 2009.

[13] Y. Wang, Z. Cai, Y. Zhou and W. Zeng, "An adaptive tradeoff model for constrained evolutionary optimization," *IEEE Tran. Evol. Comput.*, Vol. 12, no. 1, pp. 80-92, Feb. 2008.

[14] F. Peng, K. Tang, G. Chen and X. Yao, "Population-based Algorithm Portfolios for Numerical Optimization," *IEEE Tran. Evol. Comput.*, Vol. 14, no. 5, pp. 782-800, Oct. 2010.

[15] Z. Yang, K. Tang and X. Yao, "Large scale evolutionary optimization using cooperative coevolution,"*Inform. Sci.*, Vol. 178, pp. 2986-2999, Aug. 2008.

[16] X. Li and X. Yao, "Tackling high dimensional non-separable optimization problems by cooperatively coevolving particle swarms," in *Proc. IEEE CEC*, May 2009, pp. 1546-1553.

[17] X. Li and X. Yao, "Cooperatively Coevolving Particle Swarms for Large Scale Optimization," *IEEE Tran. Evol. Comput.*, accepted on 12 January 2011.

[18] F. V. den Bergh and A. Engelbrecht, "A cooperative approach to particle swarm optimization," *IEEE Tran. Evol. Comput.*, Vol. 8, no. 3, pp. 225-239, Jun. 2004.

[19] http://www.ntu.edu.sg/home/epnsugan/

[20] R.Mallipeddi, S. Mallipeddi and P. N. Suganthan, "Ensemble strategies with adaptive evolutionary programming," *Inform. Sci.*, Vol. 180, pp. 1571-1581, Jan. 2010.

[21] Q. Pan, P. N. Suganthan and M.F. Tasgetiren, "A harmony search algorithm with ensemble of parameter sets," in *Proc. IEEE CEC*, May 2009, pp. 1815-1820.

[22] R.Mallipeddi and P. N.Suganthan, "Ensemble of constraint handling techniques," *IEEE Tran. Evol. Comput.*, Vol. 14, no. 4, pp. 561-579, Aug. 2010.

[23] E.L.Yu and P. N. Suganthan, "Ensemble of niching algorithms," *Inform. Sci.*, Vol. 180, pp. 2815-2833, Apr. 2010.

[24] Z. He and G. G. Yen, "An ensemble method for performance metrics in multiobjective evolutionary algorithms," in *Proc. IEEE CEC*, Jun. 2011, pp. 1724-1729.

[25] E. Tsang and A. Kwan, "Mapping constraint satisfaction problems to algorithms and heuristics", Technical Report CSM-198, Dept. Comput. Sci., Univ. Essex, Colchester, UK, 1993.

[26] C. Si, W. Li, L. Wang and X. Wang, "A philosophy illustration of cultural algorithm and particle swarm optimization," in *Proc.8th World Cong. Intell. Contr. Auto.(WCICA)*, Jinan, China, 2010, pp. 350-355.

[27] R. Akbari and K. Ziarati, "A rank based particle swarm optimization algorithm with dynamic adaptation,"*J. Comput. Appl. Math.*, Vol. 235, no. 8, pp. 2694-2714, Feb. 2011.

[28] S. Krikpatrick, C. Gelett and M.Veechi, "Optimization by simulated annealing," *Sci.*, Vol. 220, no. 4598, pp. 671-680, May. 1983.

[29] T. P. Runarsson and X. Yao, "Stochastic ranking for constrained evolutionary optimization," *IEEE Tran. Evol. Comput.*, Vol. 4, no. 3, pp. 284-294, Sep. 2000.

[30] J. J. Liang, T. P. Runarsson, E. Mezura-Montes, M. Clerc, P. N. Suganthan, C. A. Coello Coello, and K. Deb, "Problem definitions and evaluation criteria for the CEC 2006," Special Session on Constrained Real-Parameter Optimization, Nanyang Technol. Univ., Singapore, Tech. Rep., 2006.