A Constraints Scattered Memetic Algorithm for Constrained Optimization Problem

Qunxiong Zhu
College of Information Science and Technology, Beijing University of Chemical Technology, Beijing, China
zhuqx@mail.buct.edu.cn

Wenxing Xu Zhenyu Wang* Zhiqiang Geng
College of Information Science and Technology, Beijing University of Chemical Technology, Beijing, China
estellaxu@163.com rober2005@126.com gengzhiqiang@mail.buct.edu.cn

***Abstract*—In this paper, a constraints scattered memetic algorithm (CSMA), which integrates a novel constraints scattered genetic algorithm (CSGA) and the traditional interior point method, is proposed for solving constrained optimization problems. In CSGA, a constraint scattering operation, a sub-population crossover method and a new population performance evaluation mechanism are employed. The complete constraints of a problem are divided into several sub-populations and all these sub-populations are crossed after their respective evolution process. In this way, the difficulty of obtaining feasible individuals in many strong constrained conditions is well overcome. And according to the newly defined population performance index, individuals with larger population diversity are chosen for further local search. These new mechanisms are combined in CSGA and interior point method is further employed as a local search operator for exploitation. Experiments and comparisons over a set of standard test functions show that population with better performance can be generated during the iteration of CSGA and the proposed CSMA has a better solution precision at less computation cost than most of the other algorithms reported in literature.**

***Index Terms*—memetic algorithm, constraints scattered, constrained optimization, interior point method**

I. Introduction

Genetic algorithm (GA) is a stochastic search algorithm which simulates the biological evolution. Due to its advantages of generality, feasibility and global search capability, it has been widely used in machine learning, pattern recognition, industrial optimal control and biology, etc [1,2]. However, in solving practical problems, especially in the constrained nonlinear programming problem, GA is often subjected to lower accuracy and high time consumption. Thereby, how to improve the accuracy and solution efficiency is the main research focus of GA.

To improve the quality of individual solutions in the population, a type of hybrid genetic algorithms called memetic algorithm (MA) was proposed by Pablo Moscato in 1989 [3]. Based on the simulation of the process of cultural evolution, it is a marriage between a population-based global search and the heuristic local search made by each of the individuals. In many cases, MAs have been shown to be capable of finding (near-) optimum solutions [4-6]. Nevertheless, even MAs may still fall victim to either slow or premature convergence.

Generally speaking, a constrained optimization problem in engineering application can be described as follows:

$$\begin{aligned} &Min\ f(x) \\ &s.t.\quad g_i(x) \le 0,\quad i = 1,...,m \\ &\qquad\ \ h_j(x) = 0,\quad j = 1,..., p \end{aligned} \tag{1}$$

Where $f(x)$ is the objective function of the problem, m is the number of inequality constraints and p is the number of equality constraints. Generally, a maximize function can be transformed to a minimize function.

Traditional methods of dealing with constraints include penalty function method, ranking based method, special representations and operators, repair method, separation of objectives and constraints and hybrid methods [7]. These methods either deal with constraints or adjust the gene during evolution, which help the population find better feasible solutions. However, during calculation of all these methods, populations iterate under full problem constraint conditions. It is known that the more constraints there are for the problem, the harder it is for the population to generate feasible solutions, which may soon dominate the whole population, just to result in the reduction of population diversity and deterioration of the global search capability. Thus to reduce the inhibition of constraints on generating feasible solutions during evolution could probably makes more feasible solutions and better population diversity.

In this paper, inspired by the concept of environmental stress in nature, a constraints scattered method is proposed to reduce resistance of constraints on generating feasible solutions, which improves population diversity while ensuring the generation of feasible solutions. Moreover, since this technique of providing diversiform feasible solutions mainly focuses on the global search

doi:10.4304/jcp.7.11.2788-2796

ability, the interior point method which could accelerate the constrained local search is further introduced into the approach in order to improve the algorithm accuracy. Thus, a constraints scattered memetic algorithm is proposed for constrained optimization problem to enhance the global exploration ability while ensuring local exploitation by introducing interior point method into genetic local search in the meanwhile.

The remainder of the paper is structured as follows. This paper will first analyze the reason and strategy for scattering constraints and integrating interior point method and put forward a new indicator for evaluating population performance. Then the new interior point method based constraints scattered memetic algorithm (CSMA) will be proposed. Subsequently in the next section, a series of experiments will be carried on, not only to verify the effectiveness of constraint scattering operation on improving population diversity in genetic algorithm, but also to demonstrate the efficiency of the new CSMA on deriving more accurate optimal results with less time consumption. Lastly, the conclusions will be drawn.

II. Interior Point Method Based Constraints Scattered Memetic Algorithm

A. Constraint scattering Method

In the study of ecology, many researches believed that population diversity has a significant contact with natural environment. For example, water is one of the important affecting factors [8]. Water-rich areas always mean larger diversity of plant. The population diversity appears to be large in the place which is rich in water such as the tropical rain forest, and the diversity appears to be small in desert. In another word, the survival pressure from natural environment (i.e. whether the current environment is suitable for living beings to survive) has a considerable impact on the diversity of the population. Hostile environment highlights the superiority population in species, but in a comfortable environment, relatively more species can be multiplied. Analogy to the genetic algorithm, the number of constraints stands for the pressure from natural environment. If a population is to be evolved in a model with more constraint equations, then the diversity of this population will be greatly reduced, which will reduce the global search capabilities of genetic algorithm. If a large number of constraints are scattered, multiple sub-environments will be formed. In each of these sub-environments new sub-populations, for each of which the diversity is to be developed sufficiently, can be produced and thus to improve the global search capability of the algorithm. Each sub-population evolved to certain iterations represents a set of individuals which meet part of the original constraint functions. Thus to cross these sub-populations and combine the advantages of each sub-population can produce individuals which meet the whole constraint conditions faster.

B. Crossover of Sub-populations

Each of the multiple sub-populations produced through the method above occupied a certain sub-region of the whole searching space. As shown in Fig. 1, A, B, C, D represent four sub-populations produced under different constraint conditions. Under weak constraint conditions, there may be feasible solution individuals which meet complete constraints generated in sub-populations. However, since our study is mainly about the influences of infeasible solutions on generation of feasible solutions, only the infeasible individuals in the sub-populations are represented in the following figure. E is on behalf of the possible new population that may be generated by the crossover of four sub-populations. The feasible region of objective function with two variables is a plane, with three variables or more is a hyperplane. Crossing individuals from two sub-populations is possibly to get a new individual on the connecting line of the two sub-regions [9]. The figure shows that there is a chance for the new individual on behalf of the solution to achieve the feasible region. For the traditional single-population evolution, due to the difficulty in guaranteeing the number of outstanding individuals (feasible solutions) initialized randomly and the overwhelming superiority of these outstanding individuals in evolution, the population diversity reduces sharply along with iteration, which leads to the premature of population. However, through the crossover of several sub-species, the number of outstanding individuals generated during iteration is significantly larger than the former. More importantly, because each sub-population iterates in the different constraint conditions, the diversity of new individuals generated through crossover in between is also higher.

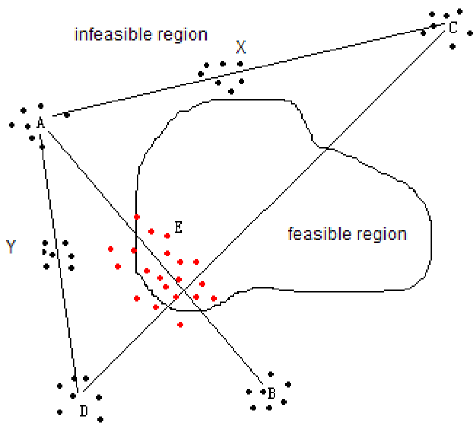

Figure 1. Crossover between sub-populations.

C. Population Performance Evaluation

In order to investigate the influence of constrains scattered method on the ratio of number of feasible solutions to population size and the diversity of feasible solutions, a new population performance evaluation index is proposed as follows.

Assumed condition: the population size is N, individual length is L, and in the t^{th} iteration, feasible solution set is $X_t = \left\{ x_i = [x_i^{t(1)} x_i^{t(2)} \cdots x_i^{t(L)}] \middle| i \in \{1,2,\cdots,K^t\} \right\}$.

Lemma 1: At the moment, the Ratio of feasible solution number to the population size $RFSN = \frac{K^t}{N}$, the average feasible solution individual can be calculated as $\overline{x}_t = E(X_t) = [\overline{x}_t^{(1)} \overline{x}_t^{(2)} \cdots \overline{x}_t^{(L)}]$, where $\overline{x}_t^{(l)} = \sum_{i=1}^{K^t} x_t^{(l)} / K^t$, and the variance of feasible solutions [10] is $D_t = E\left([X_t - E(X_t)]^2\right) = [D_t^{(1)} D_t^{(2)} \cdots D_t^{(L)}]$, where $D_t^{(l)} = \sum_{i=1}^{K^t} (x_i^{t(l)} - \overline{x}_t^{(l)})^2 / K^t$, $l \in \{1,2,\cdots,L\}$. The larger the variance is, the better the diversity of feasible solutions is.

Definition 1: FSD which represent the diversity of feasible solutions is defined as follows:

$$FSD = \|D_t\| = \sqrt{\sum_{l=1}^{L} \left(\sum_{i=1}^{K^t} (x_i^{t(l)} - \overline{x}_t^{(l)})^2 \right)^2} / K^t . \quad (2)$$

Definition 2: PP which represents the population performance index is defined as follows:

$$\begin{aligned} PP &= RFSN \bullet FSD \\ &= \frac{K^t}{N} \frac{\sqrt{\sum_{l=1}^{L} \left(\sum_{i=1}^{K^t} (x_i^{t(l)} - \overline{x}_t^{(l)})^2 \right)^2}}{K^t} . \\ &= \sqrt{\sum_{l=1}^{L} \left(\sum_{i=1}^{K^t} (x_i^{t(l)} - \overline{x}_t^{(l)})^2 \right)^2} / N \end{aligned} \quad (3)$$

The larger PP is, the better the population performance is, which means larger probability for the population to get the global optimal feasible solution.

D. Interior Point Method based Constraints Scattered Memetic Algorithm (CSMA)

Hybrid with the constraints scattered operation and crossover of sub-populations mentioned above, a new hybrid algorithm of CSGA is formed. The CSGA is supposed to be more promising in getting global feasible solutions with larger diversity. However, it still suffers from the drawback of slow convergence in the later iterations of genetic algorithm. Since MA which applies local search operators to improve the quality of individual solutions in the population can improve the searching efficiency of genetic algorithm greatly, it is considered to replace the genetic algorithm in CSGA. For constrained minimization, a suitable local search should be applied in the MA. Interior point method, which solves a sequence of approximate minimization problems derived by penalty function method by a series of attempts of Newton or conjugate gradient steps, represents the state of art in constrained minimization. Though it has been successfully applied in many scientific and engineering constrained optimization problems in the last several decades, it has the obvious drawbacks of dependence on initial feasible solutions and tendency to fall into local optima. Therefore, the interior point method is here used as a special operator to solve the local search in this paper. And together with the CSGA, which is supposed to provide various feasible individuals to serve as the initial points for interior point method, the interior point method based constraints scattered memetic algorithm (CSMA) is formed. The specific method is that we apply interior point method as the local search to each feasible individual after crossover of sub-populations derived after iterations of constraints scattered genetic algorithm. In this way, the local search for every feasible individual can help us to find a better solution.

It is worth to notice that, not all the feasible individuals would be selected to run the interior point method search for the consideration of computational cost, and neither may there be enough candidate feasible individuals for selection in some strong constrained problems. In this case, a repair method proposed by Chootinan [11] is introduced here to add additional feasible individuals. In this algorithm, the gradient information derived from the constraints is use to fix the infeasible individuals, then more feasible solutions can be generated.

Thus, the proposed CSMA is formed through combining the constraints scattered method, crossover of sub-populations and interior point method. The step of algorithm is described as follows:

Step 1: Initial the population and set the parameters (population size, crossover and mutation rate).

Step 2: Divide the population into several sub-populations with scattered constraints.

Step 3: In each sub-population use penalty function method to deal with constraints and use the individuals to perform crossover and mutation to generate the offspring.

Step 4: Cross all the sub-populations and reserve all the feasible individuals.

Step 5: Choose a certain number of feasible individuals to do the local search. If the number of feasible solution individuals is less than the set point, use repair method to generate additional feasible individuals.

Step 6: Export the global best feasible solution if the termination criteria are met, otherwise go to **step 2**.

III. Experiment And Comparison

A. Comparison of CSGA and Standard Single-population Genetic Algorithm(SGA)

For comparing the constraints scattered genetic algorithm with the standard genetic algorithm, the standard test function g04 is taken. In CSGA, the complete constrains are scattered into 3 sub-population, each sub-population has two constrains. The population size is 120 and each sub-population size is 40. 10 is set as the iteration number for sub-population evolution. Other parameters are set the same in two algorithms as follows: the crossover rate is 0.5, mutation rate is 0.05, and arithmetical crossover and uniform mutation are taken. The crossover and mutation operators are described as follows:

Arithmetical crossover:

$$\begin{cases} x_A^{t+1} = \theta x_B^t + (1-\theta) x_A^t \\ x_B^{t+1} = \theta x_A^t + (1-\theta) x_B^t \end{cases} \quad (4)$$

Where x_A^t and x_B^t are the parent individuals, x_A^{t+1} and x_B^{t+1} are the offspring, $\theta \in [0,1]$ is a constant value which derived empirically as 0.5.

Uniform mutation:

Assumed $x = \{x_1, x_2, \cdots, x_k, \cdots, x_n\}$ is the mutating individual, for a random integer $k \in [1, n]$, the mutant x' is:

$$x' = \{x_1, x_2, \cdots, x_k', \cdots, x_n\} \quad (5)$$

$$x_k' = x_{k,\min} + \gamma (x_{k,\max} - x_{k,\min}) \quad (6)$$

Where $x_{k,\min}$ and $x_{k,\max}$ are the bounds of variable x_k, $\gamma \in [0,1]$ is a random variable.

Fifty independent experiments are taken and the results of 20 runs randomly selected among them are shown in Fig. 2.

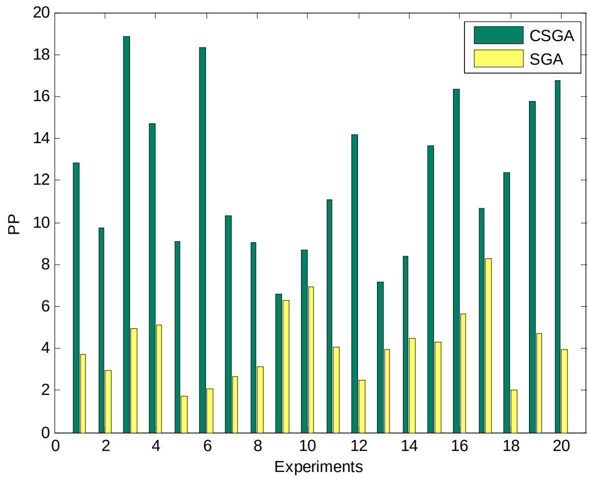

Figure 2. Population performance comparison of CSGA and SGA for g04 of 20 independent runs.

Obviously, the PP of CSGA is much larger than that of SGA, meaning a better population performance. For CSGA, the average feasible solution number is 64.89, and the diversity of feasible solutions is 22.0339, comparing with 48.65 and 10.59 of the standard GA. From this experiment we can see CSGA can get much more number of feasible solution which will benefit the interior point method to find the globe optimal.

Another experiment was carried out for analyzing the performance of PP versus iteration. All parameters are set as the same as in the first experiment except the iteration value is set equal to 300. The PP values during iterations are shown in Fig. 3.

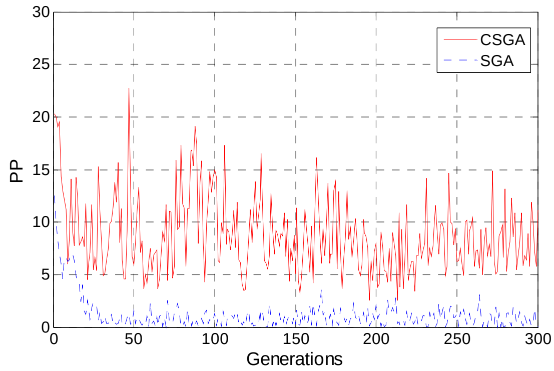

Figure 3. Evolution comparison of CSGA and SGA for g04 in a specific run

As shown in Fig. 3, the PP values in both algorithms are decreasing during the iterations, and PP of CSGA converged to about 10 after 25 generations. But as we can see, PP of SGA decreased sharply to 2 with convergence, which is apparently less than that of CSGA. Furthermore, the values of PP have a large oscillation than normal algorithm mainly due to the variety individuals of different sub-population. But it does not affect the performance of the whole algorithm, which is acceptable in the later experiments.

For the investigation of the effect on constrained optimization problem with less constrains, another experiment are carried out. In this experiment, all parameters are set as in the above experiment except the test function is replaced by g06, which only has two constraints. In CSGA, the two constraints are divided into two sub-populations. Fifty independent experiments are taken and the results of 20 runs randomly selected among them are shown in Fig. 4 and the detail data are shown in Table I.

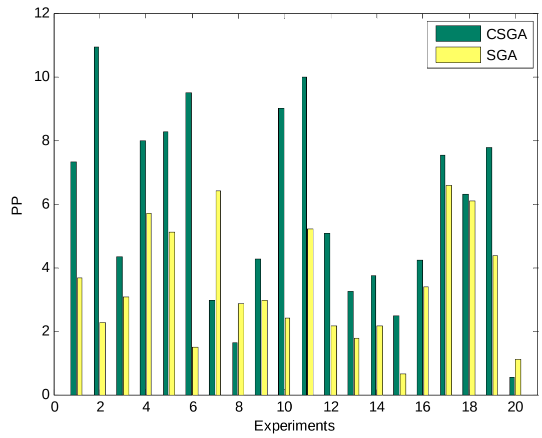

Figure 4. Population performance comparison of CSGA and SGA for g06 of 20 independent runs.

From Fig. 4, the PP of CSGA is still larger than SGA. But with a deeper analysis from Table I we can see that the feasible solution number generated by two methods are nearly the same, compared 36.1000 with 33.4500. The reason for a larger PP derived by CSGA is that the diversity of feasible solutions obtained by CSGA is

20.3942, which is larger than 11.6416 obtained SGA, which confirmed the supposition in our previous analysis.

TABLE I.
COMPARISON OF CSGA AND SGA FOR G06

Methods		SGA	CSGA
FSN	Mean	36.1000	33.4500
	St.dev	13.1665	12.3095
FSD	Mean	11.6416	20.3942
	St.dev	4.2310	6.4611
PP	Mean	3.4706	5.8526
	St.dev	1.8330	2.9984

B. Test on Constraint Allocation Scheme

In the study of CSGA, it is found that the way to scatter the constraints of a problem has a great impact on the result. In most circumstances, feasible solution can be generated by crossover operation of two infeasible solutions only if they are on different sides of the feasible area. If two sub-populations are on the same side of a feasible area, the crossover operation can hardly generate a feasible solution. In Fig. 1, for instance, area A and B or area C and D are respectively on both sides of the feasible area, the solution generated by them can reach the feasible area most probably. On the contrary, area A and C or area A and D are on the same side of the feasible area, and then there is little chance to find a feasible solution according to our analysis.

It can be seen that with the same initial population size and other parameters, a larger number of sub-population(less sub-population size) means a greater chance to generate an effective crossover, but small sub-population size limit the number of feasible solutions generated. Meanwhile, a larger sub-population size (fewer sub-populations) can generate more feasible solution if the sub-populations are on both sides of the feasible area, but there will be much fewer feasible solutions if they are on the same side of the feasible area.

In order to achieve a balance between sub-population size and number, the allocation scheme of constraints is tested here. In this experiment, we use three different ways of constrains scatter for g04. Type-1 is that six constrains of g04 are divided into six sub-populations; Type-2 is that constrains are divided into three sub-populations, each one includes two constrains; Type-3 means that constrains are divided into two sub-populations and each one gains three constrains. Fifty independent experiments are taken and results are shown in Fig. 5 and Table II.

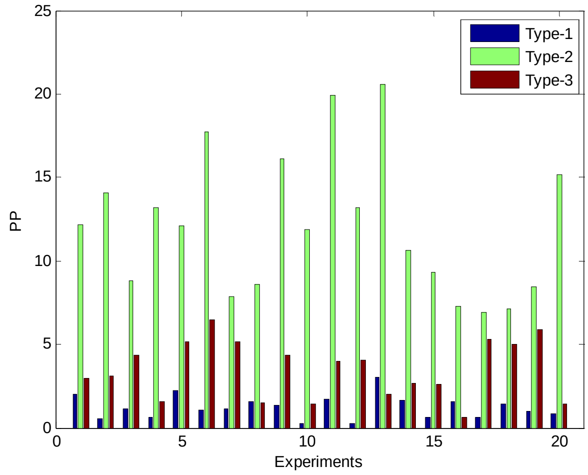

Figure 5. Population performance comparison of three constraints allocation schemes for g04.

TABLE II.
COMPARISON OF CSGA WITH THREE CONSTRAINTS ALLOCATION SCHEMES AND SGA FOR G06

Methods		SGA	CSGA		
			Type-1	*Type-2*	*Type-3*
FSN	Mean	48.6500	12.5000	65.3000	17.5000
	St.dev	13.3775	6.2761	8.2199	7.4125
FSD	Mean	10.5922	19.6936	21.0025	21.8417
	St.dev	3.4522	8.1886	5.8419	7.3703
PP	Mean	4.1916	1.9891	11.4458	3.5198
	St.dev	1.4225	1.1124	3.7442	2.6138

Fig. 5 shows population performance comparison of three constraint allocation schemes of 20 runs in a histogram form. It is obvious that Type-2 achieves the best results in three types, the average PP of which is 11.4458, while the average PP of the Type-1 is 1.9891 and of Type-3 is 3.5198. Further from Table II, it can be seen that a proper way of constrains scattering has a significant effect on feasible solution searching, and if is chosen correctly, such as Type-2, a better population performance than standard genetic algorithm can be derived.

We can also see that if we can set the sub-population exactly on the both sides of the feasible area, then the effect of each crossover operation are guaranteed, leading to a better feasible solution searching. But in this paper we haven't do enough research on this problem and all of the constraints allocation schemes are determined by empirical method.

C. Experiments on the Influences of Population Size on Population Performance

In this section we did a series of experiments on the effect of sub-population size on feasible solution searching. Constraints allocation scheme Type-2 is chosen and the whole population size are set from 120 to 360(sub-population size is from 40 to 120). Fifty independent experiments are taken and the results are shown in Fig. 6 and Table III.

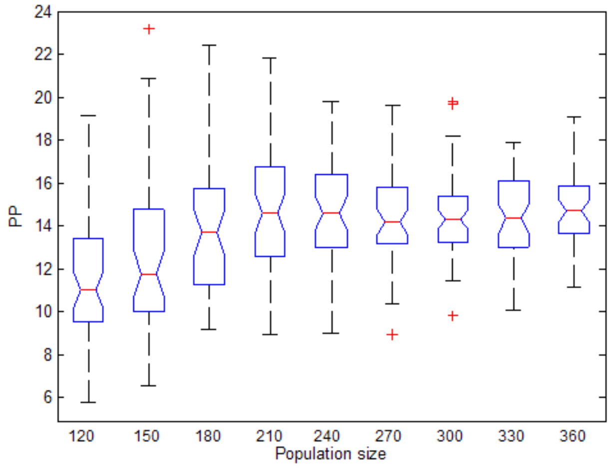

Figure 6. Variation of population performance versus population size.

TABLE III.
COMPARISON OF CSGA WITH THREE CONSTRAINTS ALLOCATION SCHEMES AND SGA FOR G06

popsize	Results of PP				FEs
	Max	Min	Mean	St.dev	
120	19.1531	5.7875	11.4421	3.2209	1200
150	23.1774	6.5296	12.7613	4.0489	1500
180	22.4325	9.1632	13.8296	2.9906	1800
210	21.8330	8.9780	14.4948	3.0542	2100
240	19.8600	9.0229	14.8200	2.4167	2400
270	19.6035	8.9569	14.5767	2.2110	2700
300	19.8034	9.8058	14.3992	1.9236	3000
330	17.8805	10.0585	14.5472	1.9921	3300
360	19.0649	11.1347	14.5472	1.8091	3600

From Fig. 6, the value of PP increases with the increasing population size, till the population size reaches about 240, when the value of PP stop increasing and stabilizes to about 15. Also the stability of population performance is getting better with the increase of population size. It proves that the greater the population of a species, the greater the chance to survive. But as in genetic algorithm, computational cost is considered as a performance evaluation criterion, we should choose a proper population size to improve the algorithm's performance while ensuring the acceptance of computational cost as well.

D. Experiments on Standard Test Functions

In order to prove the feasibility and validity of the proposed method, the comparisons are taken among the three state-of-the-art approaches that are Simple Multi-membered Evolution Strategy (SMES) [12], Self Adaptive Penalty Function (SAPF) [13], and Cultured differential evolution (CULDE) [14]. The standard test function g01, g04, g07, g10 are taken because they have relatively more constraints. These test functions are listed in Appendix A. All of their constraints are divided into three sub-populations. The parameters are set as follows: popsize is 120, crossover rate is 0.6, mutation rate is 0.05. 70 feasible individuals are selected from the sub-population crossover. And the arithmetical crossover and uniform mutation are taken.

The value of function evaluations (FEs) is an important evaluation criterion for the computational cost. Less FEs indicate higher solution efficiency. Fifty independent experiments are taken for each test function. Comparing with the current three state-of-the-art approaches, the results are shown in Table IV.

TABLE IV.
COMPARISON OF CSMA WITH RESPECT TO SMES, SAPF AND CULDE

Function	Method	Best known	Best	Mean	Std	Worst	AvgFEs
g01	CSMA	-15	**-15.000**	-14.999	2.0996E-4	-14.999	2474
	SMES		-14.999	-14.960	2.10E-1	-13.828	240000
	SAPF		-15.000	-14.552	0.700	-13.097	500000
	CULDE		-15.000000	-14.999996	0.000002	-14.999993	100100
g04	CSMA	-30665.539	**-30665.539**	-30665.538	6.8704E-4	-30665.537	4642
	SMES		-30665.539	-30665.531	1.35E-2	-30665.473	240000
	SAPF		-30665.401	-30665.9221	2.043	-30656.471	500000
	CULDE		-30665.53867	-30665.53867	0.000000	-30665.53867	100100
g07	CSMA	24.306209	24.306221	24.306301	1.0594e-4	24.306495	15956
	SMES		24.473	24.734	2.15E-1	25.401	240000
	SAPF		24.838	27.328	2.172	33.095	500000
	CULDE		24.306209	24.30621	1E-6	24.306212	100100
g10	CSMA	7049.25	**7049.248**	7050.484	1.5023	7058.762	51841
	SMES		7076.725	7330.398	153.72E+0	7816.830	240000
	SAPF		7069.981	7238.964	137.773	7489.406	500000
	CULDE		7049.248058	7049.248266	0.000167	7049.248480	100100

In the problem g01, the best result obtained by our approach was f(x) =-15.000, with x={1.000, 1.000, 1.000, 1.000, 1.000, 1.000, 1.000, 1.000, 1.000, 3.000, 3.000, 3.000, 1.000} and the values $g_i(x)$ ={0.000, 0.000, 0.000, -5.000, -5.000, -5.000, 0.000, 0.000, 0.000} for the constraints. SMES and SAPF can also reach the optimum, however they were not very robust in this problem, for the mean and worst value performed not well enough than ours. CULDE had competitive results of best mean and worst value compared with CSMA, and the standard deviation value is even better than ours.

G04 is a problem which is easy to solve, The best result obtained for this problem is f(x) =-30665.539, with x={1.000, 1.000, 1.000, 1.000, 1.000, 1.000, 1.000, 1.000, 1.000, 3.000, 3.000, 3.000, 1.000} and $g_i(x)$ = {0.000, 0.000, 0.000, -5.000, -5.000, -5.000, 0.000, 0.000, 0.000} for the constraints. SMES and CULDE can also find the optimal solution but SAPF failed. Our CSMA can obtain a value very close to the optimum in all runs, which is better than SMES and SAPF. CULDE had the almost same result with ours.

The best result obtained by CSMA in g07 is f(x) =24.306221, with x= {2.1719970, 2.363683, 8.773922, 5.0959875, 0.990655, 1.430576, 1.321644, 9.8287253, 8.2800905, 8.3759219} and $g_i(x)$ = {-0.000001, 0.000009, -0.000001, -0.000152, -0.000013, -0.000009, -6.148508, -50.023938}. Only CULDE can find the optimum in this problem, and both SMES and SAPF were failed. Our method obtained a value which is very close to the optimum and was more robust than SMES and SAPF.

In g10, our approach can reach the optimal solution f(x) =7049.248, with x= {579.310, 1359.990, 5109.947, 182.018, 295.602, 217.982, 286.416, 395.602} and the values $g_i(x)$ = {0.000, 0.000, 0.000, 0.033, -0.236, 0.300} for the constraints. Again, SMES and SAPF failed to find the optimum, and their robust were also not very well. CULDE obtained competitive results in best, mean and standard deviation value and had a worst value which was better than ours.

The results of SMES were obtained with 240000 evaluations of the fitness function, the results of SAPF required 500000 evaluations, and the results of CULDE required 100100 evaluations of the fitness function. ISHGA only need 66000 FEs, which was apparently much less than the other three techniques.

For a more intuitive observation of the convergence process of our new proposed method, the evolution curve of CSMA for g04 in a specific run is compared with that of SGA and the result is shown in Fig. 7.

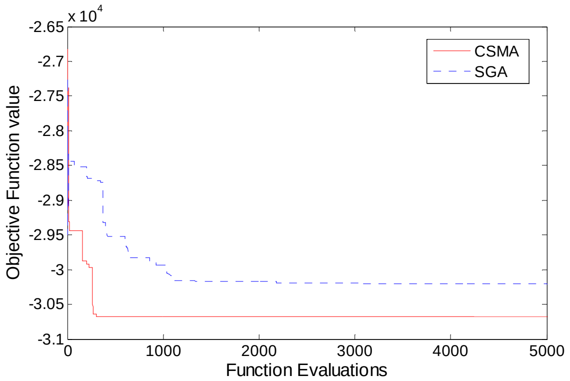

Figure 7. Evolution comparison of CSMA and SGA for g04.

In Fig. 7, the optimal solution (-30665.539) obtained by CSMA is much better than the one (-3.02e+4) obtained by SGA. Meanwhile the CSMA has an outstanding converge rate (converged at about 300 FEs), which is better than SGA (converged at about 1000 FEs), showing that the combination of CSGA and interior point method is successful.

In short, our CSMA approach performs well in solution precision and efficiency, which is better than SMES and SAPF. CULDE has the almost same solution ability but the computation cost is much larger than our method.

IV. CONCLUSIONS

In this paper, a constraint scattering method is employed in memetic algorithm to construct a novel constraints scattered memetic algorithm (CSMA) for solving constrained optimization problems. Three new mechanisms are combined in the new constraints scattered genetic algorithm (CSGA). In our proposed CSGA, the individuals evolve in a relatively weak constrained environment to gain more feasible solutions with larger diversity in each sub-population. In order to assess the effectiveness of this strategy, a new population performance evaluation index is defined, and experiments have shown that CSGA can get much more number of feasible solutions with larger population diversity. Further, constraint allocation schemes are discussed to help enlarge the whole population feasible solution number and diversity. Moreover, after a selection of feasible individuals, the interior point method is used here as the local search operator of the memetic algorithm to replace the genetic algorithm in CSGA especially for constrained optimization problems. The results of experiments on four standard test functions demonstrate that our method can achieve a better optimal solution with less computational cost (measured in FEs). Our future work is to improve the robustness of this algorithm which means to reduce the dependence on parameter selection for different problems.

APPENDIX A TEST FUNCTIONS

A.1. g01

Minimize $f(\vec{x}) = 5\sum_{i=1}^{4} x_i - 5\sum_{i=1}^{4} x_i^2 - \sum_{i=5}^{13} x_i$

s.t

$$g_1(\vec{x}) = 2x_1 + 2x_2 + x_{10} + x_{11} - 10 \le 0,$$

$$g_2(\vec{x}) = 2x_1 + 2x_3 + x_{10} + x_{12} - 10 \le 0,$$

$$g_3(\vec{x}) = 2x_2 + 2x_3 + x_{11} + x_{12} - 10 \le 0,$$

$$g_4(\vec{x}) = -8x_1 + x_{10} \le 0,$$

$$g_5(\vec{x}) = -8x_2 + x_{11} \le 0,$$

$$g_6(\vec{x}) = -8x_3 + x_{12} \le 0,$$

$$g_7(\vec{x}) = 2x_4 - x_5 + x_{10} \le 0,$$

$$g_8(\vec{x}) = 2x_6 - x_7 + x_{11} \le 0,$$

$$g_9(\vec{x}) = 2x_8 - x_9 + x_{12} \le 0,$$

where $0 \le x_i \le 1 (i = 1, \cdots, 9)$, $0 \le x_i \le 100 (i = 10,11,12)$, $0 \le x_{13} \le 1$.

The optimum solution is $x^* = (1,1,1,1,1,1,1,1,1,3,3,3,1)$ where $f(x^*) = -15$.

A.2. g04

Minimize $f(\vec{x}) = 5.3578547x_3^2 + 0.8356891x_1x_5 + 37.293239x_1 - 40792.141$

s.t.

$g_1(\vec{x}) = 85.334407 + 0.0056858x_2x_5 + 0.0006262x_1x_4 - 0.0022053x_3x_5 - 92 \le 0,$

$g_2(\vec{x}) = -85.334407 - 0.0056858x_2x_5 - 0.0006262x_1x_4 + 0.0022053x_3x_5 \le 0,$

$g_3(\vec{x}) = 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 + 0.0021813x_3^2 - 110 \le 0,$

$g_4(\vec{x}) = -80.51249 - 0.0071317x_2x_5 - 0.0029955x_1x_2 - 0.0021813x_3^2 + 90 \le 0,$

$g_5(\vec{x}) = 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 + 0.0019085x_3x_4 - 25 \le 0,$

$g_6(\vec{x}) = -9.300961 - 0.0047026x_3x_5 - 0.0012547x_1x_3 - 0.0019085x_3x_4 + 20 \le 0,$

where $78 \le x_1 \le 102, 33 \le x_2 \le 45, 27 \le x_i \le 45 (i = 3,4,5)$. The optimum solution is $x^* = (78,33,29.995256025682,45,33.775812905788)$, where $f(x^*) = -30665.539$

A.3. g06

Minimize $f(\vec{x}) = (x_1 - 10)^3 + (x_2 - 20)^3$

s.t.

$g_1(\vec{x}) = -(x_1 - 5)^2 - (x_2 - 5)^2 + 100 \le 0,$

$g_2(\vec{x}) = -(x_1 - 6)^2 + (x_2 - 5)^2 - 82.81 \le 0,$

where $13 \le x_1 \le 100, 0 \le x_2 \le 100$. The optimum solution is $x^* = (14.095, 0.84296)$, where $f(x^*) = -6961.81388$.

A.4. g07

Minimize

$f(\vec{x}) = x_1^2 + x_2^2 + x_1x_2 - 14x_1 - 16x_2 + (x_3 - 10)^2 + 4(x_4 - 5)^2 + (x_5 - 3)^2 + 2(x_6 - 1)^2 + 5x_7^2 + 7(x_8 - 11)^2 + 2(x_9 - 10)^2 + (x_{10} - 7)^2 + 45$

s.t.

$g_1(\vec{x}) = -105 + 4x_1 + 5x_2 - 3x_7 + 9x_8 \le 0,$

$g_2(\vec{x}) = 10x_1 - 8x_2 - 7x_7 + 2x_8 \le 0,$

$g_3(\vec{x}) = -8x_1 + 2x_2 + 5x_9 - 2x_{10} - 12 \le 0,$

$g_4(\vec{x}) = 3(x_1 - 2)^2 + 4(x_2 - 3)^2 + 2x_3^2 - 7x_4 - 120 \le 0,$

$g_5(\vec{x}) = 5x_1^2 + 8x_2 + (x_3 - 6)^2 - 2x_4 - 40 \le 0,$

$g_6(\vec{x}) = x_1^2 + 2(x_2 - 2)^2 - 2x_1x_2 + 14x_5 - 6x_6 \le 0,$

$g_7(\vec{x}) = 0.5(x_1 - 8)^2 + 2(x_2 - 4)^2 + 3x_5^2 - x_6 - 30 \le 0,$

$g_8(\vec{x}) = -3x_1 + 6x_2 + 12(x_9 - 8)^2 - 7x_{10} \le 0,$

where $-10 \le x_i \le 10 (i = 1, \cdots, 10)$. The optimum solution is $x^* = (2.171996, 2.363683, 8.773926, 5.095984, 1.430574, 1.321644, 9.828726, 8.280092, 8.375927)$ where $f(x^*) = 24.3062091$.

A.5. g10

Minimize $f(\vec{x}) = x_1 + x_2 + x_3$

s.t.

$g_1(\vec{x}) = -1 + 0.0025(x_4 + x_6)^2 \le 0,$

$g_2(\vec{x}) = -1 + 0.0025(x_5 + x_7 - x_4)^2 \le 0,$

$g_3(\vec{x}) = -1 + 0.01(x_8 - x_5)^2 \le 0,$

$g_4(\vec{x}) = -x_1x_6 + 833.33252x_4 + 100x_1 - 83333.333 \le 0,$

$g_5(\vec{x}) = -x_2x_7 + 1250x_5 + x_2x_4 1250x_4 \le 0,$

$g_6(\vec{x}) = -x_3x_8 + 1250000 + x_3x_5 - 2500x_5 \le 0,$

where $100 \le x_1 \le 10000$, $1000 \le x_i \le 10000 (i = 2,3)$, $10 \le x_i \le 100 (i = 4, \cdots, 8)$.

The optimum solution is $x^* = (579.19, 1360.13, 5109.92, 182.0174, 295.5985, 217.9799, 286.40, 395.5979)$

where $f(x^*) = 7049.25$

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China (Grants No. 61074153), and the Fundamental Research Funds for the Central Universities (Grants No. zz1136).

REFERENCES

[1] B. F. David, "An introduction to simulated evolutionary optimization," IEEE Transaction on Neural Networks,vol. 5, Jan. 1994, pp.3-14.

[2] X. Yao, G. L. Chen, H. M. Xu, "A survey of evolutionary algorithms," Chinese Journal of Computers, vol. 18, pp. 694-706, Sept. 1995.

[3] P. Moscato. “On evolution, search, optimization, genetic algorithms and martial arts: towards memetic algorithms,” Pasadena, California, USA: Technical Report. Caltech Concurrent Computation Program, Report 826, California Institute of Technology, 1989.
[4] M. D. Liu, “The Development of the Memetic Algorithm,” Control Theory and Applications, vol. 26, pp. 1-5, 2007.
[5] D. K. Dimitroulas, P. S. Georgilakis, “A new memetic algorithm approach for the price based unit commitment problem,” Applied Energy, Vol. 88, pp. 4687-4699, December 2011.
[6] P. C. Pendharkar, “A multi-agent memetic algorithm approach for distributed object allocation,” Journal of Computational Science, in Press.
[7] C. A. C. Coello, “Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art,” Computer methods in applied mechanics and engineering, vol.191, 2002, pp.1245-1287.
[8] A. H. Gentry, Changes in plant community diversity and floristic composition on environmental and geographical gradients, Annal of the Missouri Botanical Garden, 1988, 75:1-34.
[9] Michalewicz, “Evolutionary algorithms for constrained engineering problems,” IWEC’2000 International Workshop on Evolutionary Computation, Wuhan, 2000, pp.1-11.
[10] X. H. Zhang, G. Z. Dai, N. P. Xu, “Study on Diversity of Population in Genetic Algorithms,” Control theory and applications, vol.15, Feb. 1998, pp.17-22.
[11] P. Chootinan, A. Chen, “Constraint handling in genetic algorithms using a gradient-based repair method,” Computers and Operations Research, vol.33, 2006, pp.2263-2281.
[12] E. Mezura-Montes, C. A. C. Coello, “A Simple Multi-membered evolution strategy to solve constrained optimization problems,” IEEE Transactions on Evolutionary Computation, vol.9, Feb 2005, pp.1-17.
[13] B. Tessema, G. G. Yen, “A self adaptive penalty function based algorithm for constrained optimization,” IEEE Congress on Evolutionary Computation, Canada, 2006, pp.246-253.
[14] R. L. Becerra, C. A. C. Coello, “Cultured differential evolution for constrained optimization,” Computer Methods in Applied Mechanics and Engineering, vol.195, 2006, pp.4303-4322.

Qunxiong Zhu 1960-, Ph.D, Professor, Dean of College of Information Science and Technology in Beijing University of Chemical Technology. Area of research: Artificial Intelligence, Fault detection, Data Mining, Decision-making and Control research area. E_mail: zhuqx@mail.buct.edu.cn

Wenxing Xu 1982-, Ph.D candidate in Beijing University of Chemical Technology. Area of research: intelligent computing. E_mail: estellaxu@163.com

Zhenyu Wang 1987-, Master candidate in Beijing University of Chemical Technology. Area of research: intelligent computing.
E_mail: robert2005@126.com

Zhiqiang Geng 1973-, Ph.D, Associate Professor in Beijing University of Chemical Technology. Area of research: Artificial Intelligence, Control research area.
E_mail:gengzhiqiang@mail.buct.edu.cn.