

# History Information Based Optimization of Additively Decomposed Function with Constraints

Qingsheng Ren<sup>1</sup>, Jin Zeng<sup>2</sup>, and Feihu Qi<sup>1</sup>

<sup>1</sup> Department of Computer Science and Engineering,  
Shanghai Jiaotong University, Shanghai 200030, P.R. China  
{ren-qs, fhqi}@cs.sjtu.edu.cn

<sup>2</sup> Department of Mathematics,  
Shanghai Jiaotong University, Shanghai 200030, P.R. China  
zengjin@sjtu.edu.cn

**Abstract.** In this paper, we propose a modified estimation of distribution algorithm HCFA (History information based Constraint Factorization Algorithm) to solve the optimization problem of additively decomposed function with constraints. It is based on factorized distribution instead of penalty function and any transformation to a linear model or others. The history information is used and good results can be achieved with small population size. The feasibility of the new algorithm is also given.

## 1 Introduction

The constraint problem we discuss here is defined as the following:

$$\begin{aligned} \max f(X) &= \sum_i f_i(S_i) \\ \text{s.t. } C_i(S_i) &\end{aligned} \quad (1)$$

where  $X = \{x_1, \dots, x_n\}$  and the value of the  $i$ th variable belongs to the set  $\{x_{i,1}, \dots, x_{i,n_i}\}$ .  $C_i(S_i)$  stands for the  $i$ th constraint function (it may be equality or inequality) and the variable set  $S_i \subseteq X$  ( $i = 1, \dots, l$ ). The function  $f(x)$  is called additively decomposed function (ADF). This class of functions is of great theoretical and practical importance. Optimization of an arbitrary function in this space is NP complete.

Evolutionary computation (EC) is widely used for constraint optimization [1]. This method is loosely based on the mechanics of artificial selection and genetic recombination. Two crucial factors of the EC success, a proper growth and mixing of good building blocks, are often not achieved [2]. The problem of building block disruption is often referred to as the linkage problem [3]. Various attempts to prevent the disruption of important partial solutions have been done. One way to generate new individuals is to use the information extracted from the entire set of promising solutions. A general scheme of the algorithms based

on this principle is called the estimation of distribution algorithm (EDA)[4]. The main problems of EDA are how to estimate the distribution and how to get new individuals by this distribution. EDA is used successfully not only for unconstrained optimization but also for constrained optimization [5].

Although EDA is used successfully in real application, it also has the shortcoming of large population size. One of the reasons is that EDA only uses the information of the selected population and does not use any information of the population before selection. In this paper a history information based algorithm is proposed to solve additively decomposed function with constraints. We called this new algorithm as HCFA (History information based Constraint Factorization Algorithm). The main advantage of HCFA is that it can get good results with small population size. It can always produce feasible solutions if the individuals of the initial population are all feasible by using a factorization of the distribution of selected points. It can handle not only linear constraints but also nonlinear constraints. The paper is organized as follows. Section 2 gives the frame of the algorithm. In section 3, some important topics, including initialization and feasibility are discussed. Some numerical results are given in section 4 to show the efficiency of the algorithm.

## 2 The Frame of the Algorithm

### 2.1 Factorization

To solve the problem, we must get a factorization of the probability of distribution at first. For convenience, just suppose  $X = \bigcup_{i=1}^l S_i$ . Define

$$d_i = \bigcup_{j=1}^i S_j, b_i = S_i \setminus d_{i-1}, c_i = S_i \cap d_{i-1}$$

Set  $d_0 = \phi$  and then we get the factorization as

$$P(X) = \prod_{i=1}^l P(x_{b_i} | x_{c_i})$$

Let's see an example. Suppose  $X = \{x_1, x_2, x_3\}$ ,  $x_i \in \{0, 1\}$  and the constraint functions are:

$$\begin{aligned} x_1 + x_2 &\leq 1 \\ x_2 + x_3 &\leq 1 \end{aligned}$$

Then we have

$$S_1 = \{x_1, x_2\}, S_2 = \{x_2, x_3\}$$

So the factorization is

$$P(X) = P(x_1, x_2) P(x_3 | x_2)$$

If

$$b_i \neq \phi, \forall i = 1, \dots, l; d_l = X$$

$$\forall i \geq 2, \exists j < i \text{ such that } c_i \subseteq S_j$$

we say the factorization satisfy the *running intersection property*. At this condition,  $P(x_1, \dots, x_n) = \prod_{i=1}^l P(x_{b_i} | x_{c_i})$  really holds [6].

If the running intersection property is violated the factorization might not be exact. But we will show the running intersection property is not necessary by the numerical examples.

## 2.2 Algorithm

We assume the factorization of the probability distribution is given. The following is the frame of the algorithm HCFA to solve the constraint problem:

1. Get initial feasible population;
2. Selection;
3. Compute the probabilities  $P^s(x_{b_i} | x_{c_i}, t - 1)$  using the selected individuals;
4. Generate a new population according to

$$P(x, t) = \prod_{i=1}^l [\lambda P^s(x_{b_i} | x_{c_i}, t - 1) + (1 - \lambda) P^s(x_{b_i} | x_{c_i}, t - 2)] \quad (2)$$

where  $0 < \lambda \leq 1$

5. If the termination criterion is met, finish;
6. Add the best point of the previous generation to the generated points;
7.  $t = t + 1$ , go to 2).

This algorithm can be run with any popular selection methods.

## 3 Further Discussion

### 3.1 Feasibility

Before the further discussion, we should know if the algorithm HCFA is feasible. This means the final solution should be feasible. The following theorem gives us the guarantee.

**Theorem 1.** *If the former generation is feasible, the new generation will be feasible too.*

*Proof:* If  $\exists x \in \text{Population}(t)$  and  $x$  doesn't satisfy the  $k$ th constraint  $C_k(S_k)$ . Then

$$0 \neq P(x, t) = \prod_{i=1}^l [\lambda P^s(x_{b_i}|x_{c_i}, t-1) + (1-\lambda) P^s(x_{b_i}|x_{c_i}, t-2)]$$

$$\rightarrow P^s(x_{b_k}|x_{c_k}, t-1) \neq 0 \text{ or } P^s(x_{b_k}|x_{c_k}, t-2) \neq 0$$

$$\rightarrow P^s(x_{S_k}, t-1) \neq 0 \text{ or } P^s(x_{S_k}, t-2) \neq 0$$

Then  $\exists \tilde{x} \in \text{Population}(t-1)$ ,  $\tilde{x}_{S_k} = x_{S_k}$  or  $\exists \hat{x} \in \text{Population}(t-2)$ ,  $\hat{x}_{S_k} = x_{S_k}$ . Therefore  $\tilde{x}$  or  $\hat{x}$  does not satisfy  $C_k(S_k)$

And it is impossible because we suppose the former generation is feasible.

### 3.2 How to Get Feasible Initial Population

Although Theorem 1 guarantees that we can get a feasible population from a feasible former, we need a feasible initial population. The easiest way is to create a solution randomly and then check the feasibility. When  $t = 1$ , we will generate new population according to  $P(x, 1) = \prod_{i=1}^l P^s(x_{b_i}|x_{c_i}, 0)$ .

## 4 Numerical Results

In the following examples, the experiment results are expressed by how many runs it gets the global optimum among 100 independent runs. The max generation is 100 and the population size is denoted as N.

*Example 1.*

$$\begin{aligned} & \max \sum_{i=1}^n x_i \\ & \text{s.t. } 2 \leq x_{2j-1}^2 + x_{2j}^2 + x_{2j+1}^2 \leq 8 \end{aligned}$$

where  $n = 2m + 1$ ,  $x_i \in \{0, \pm 1, \pm 2\}$ ,  $i = 1, \dots, n$ ,  $j = 1, \dots, m$ .

This example is a nonlinear programming problem with concave domain. The table 1 shows the results. When  $\lambda = 1$ , it is the algorithm used in [5]. From this table we can see that the new algorithm can get better results with smaller population size.

*Example 2.*

$$\begin{aligned} & \max \sum_{i=1}^n x_i \\ & \text{s.t. } x_{2j-1} + x_{2j} + x_{2j+1} \leq 2 \\ & \quad x_1 + x_{n-1} + x_n \leq 2 \end{aligned}$$

where  $n = 2m$ ,  $x_i \in \{0, 1\}$ ,  $i = 1, \dots, n$ ,  $j = 1, \dots, m - 1$

**Table 1.** Results of Example 1

N	1000		2000		3000		4000	
$\lambda$	[5]	HCFA	[5]	HCFA	[5]	HCFA	[5]	HCFA
	1	0.5	1	0.5	1	0.5	1	0.5
n=101	0	3	46	74	80	97	97	100
n=201	0	0	0	5	14	66	55	93
n=299	0	0	0	0	5	50	37	89

**Table 2.** Results of Example 2. Not satisfy the running intersection property

N	200		500		700		1000	
$\lambda$	[5]	HCFA	[5]	HCFA	[5]	HCFA	[5]	HCFA
	1	0.5	1	0.5	1	0.5	1	0.5
n=200	1	12	81	88	99	100	99	100
n=300	0	0	45	80	79	97	99	100
n=398	0	0	41	85	88	100	98	100
n=502	0	0	8	58	60	98	95	100

The table 2 shows the results. When  $\lambda = 1$ , it is the algorithm used in [5].

This example is a linear programming problem. In order to achieve the same convergence ratio, the population size needed by the new algorithm is much smaller than that of [5]. At the same time the factorization we used is

$$P(x_1, \dots, x_n) = P(x_1x_2x_3)P(x_4x_5|x_3) \cdots P(x_{n-2}x_{n-1}|x_{n-3})P(x_n|x_1x_{n-1})$$

Obviously it does not satisfy the running intersection property. But we can still get the global optimum. We can also use the following factorization which satisfies the running intersection property.

$$\begin{aligned} &P(x_1, \dots, x_n) \\ &= P(x_1x_2x_3)P(x_4x_5|x_1x_3) \cdots P(x_{n-2}x_{n-1}|x_1x_{n-3})P(x_n|x_1x_{n-1}) \end{aligned}$$

Table 3 shows the result with the above factorization. We get much better results just because we change the factorization. This example shows the running

**Table 3.** Results of Example 2. Satisfy the running intersection property

N	200		500		700		1000	
$\lambda$	[5]	HCFA	[5]	HCFA	[5]	HCFA	[5]	HCFA
	1	0.5	1	0.5	1	0.5	1	0.5
n=200	27	78	95	100	99	100	100	100
n=300	5	34	80	98	95	100	100	100
n=398	6	51	97	99	100	100	100	100
n=502	0	27	86	99	98	100	100	100

intersection property is not necessary. In our research we do find that some examples can not converge to the global optimum just because the factorization does not satisfy the running intersection property. We still need further study.

## 5 Conclusion

In this paper we proposed a new algorithm based on factorized distribution to solve constraint optimization problems of additively decomposed function. Future research work will focus on the running intersection property, converge theory, population size, etc.

## Acknowledgement

This project was supported by the National Natural Science Foundation of China.

## References

1. Coello, C., Carlos, A.: A Survey of Constraint Handling Techniques Used with Evolutionary Algorithms. Technical Report Lania-RI-9904. Laboratorio Nacional de Informtica Avanzada, Xalapa, Veracruz, Mexico(1999)
2. Thierens, D.: Analysis and Design of Genetic Algorithms. PhD thesis. Katholieke Universiteit Leuven, Leuven, Belgium(1995)
3. Harik, G., Goldberg, D. E.: Learning Linkage. In: Richard, K., Michael, D.(eds.): Foundations of Genetic Algorithms 4. Morgan Kaufmann, San Francisco (1997) 247-262
4. Muehlenbein, H., Paaß G.: From Recombination of Genes to the Estimation of Distributions I: Binary Parameters. Lecture Notes in Computer Science, Vol. 1411. Springer-Verlag, Berlin Heidelberg New York (1996) 178-187
5. Ren, Q., Zeng, J., Qi, F.: Optimization of Additively Decomposed Function with Constraints. Journal of Engineering and Electronics. 4(2002)85-90
6. Muhlenbein, H., Mahnig, T. and Rodriguez, A.O.: Schemata, Distribution and Graphical Models in Evolutionary Optimization. Journal of Heuristics. 5(1999) 215-247