Genetic Algorithm Combination of Boolean Constraint Programming for Solving Course of Action Optimization in Influence Nets

Yanguang Zhu, Dongliang Qin, Yifan Zhu and Xingping Cao
National University of Defense Technology, Changsha, P. R. China
Email: sailor1109@tom.com

Abstract—**A military decision maker is typically confronted by the task of determining optimal course of action under some constraints in complex uncertain situation. Thus, a new class of Combinational Constraint Optimization Problem (CCOP) is formalized, that is utilized to solve this complex Operation Optimization Problem. The object function of CCOP is modeled by Influence net, and the constraints of CCOP relate to resource and collaboration. These constraints are expressed by Pseudo-Boolean and Boolean constraints. Thus CCOP holds a complex mathematical configuration, which is expressed as a 0 1 integer optimization problem with compositional constraints and unobvious optimal object function. A novel method of Genetic Algorithm (GA) combination of Boolean Constraint Programming (BCP) is proposed to solve CCOP. The constraints of CCOP can be easily reduced and transformed into Disjunctive Normal Form (DNF) by BCP. The DNF representation then can be used to drive GA so as to solve CCOP. Finally, a numerical experiment is given to demonstrate the effectiveness of above method.**

Index Terms—**course of action optimization; influence nets; boolean and pseudo-boolean constraints; genetic algorithm; boolean constraint programming**

I. INTRODUCTION

Within military operations, targeting must be focused on creating specific effects to achieve the joint force commander's objectives. A military decision maker is typically confronted by the task of determining optimal course of action under some constraints in a complex uncertain situation. The pre-requisite of this task is the modeling of cause-effect/relevance relationships among the variables existing in the environment. The use of probabilistic reasoning framework as a modeling tool for capturing such relationships has became more popular. Commonly referred to as Influence Net (IN) [1], the framework is proposed to overcome the intractability issues in Bayesian Network [2], and has been experimentally used in the area of Effects-Based Operations (EBO) [3]. Sajjad etc. have made some research on the problem of Influence nets based course of action optimization [4, 5], but constrains of which are not fully modeled. Thus, a new class of Combinational Constraint Optimization Problem (CCOP) is formalized in Influence Net, by introducing resource and collaboration constrains.

The object function of CCOP cannot be described by analytic expression, thus traditional methods are useless for solving this class of Constraint Optimization Problem (COP). Evolutionary Algorithm (EA) is rather reliable and effective than traditional methods to solve COP that does not require obvious object of COP [6, 7]. Penalty function is the most widely used technique to handle constraints [8, 9]. However, its performance is highly dependent on the definition of penalty factors. Currently, Integration of optimization and constraint programming is a novel approach for solving COP [10]. The constraint programming is usually used to eliminate part of non-feasible solutions, which is implemented either before or middle optimization implementation. Because of their complementary strengths, they are profitably merged. Thus an optimization technique that combines the Genetic Algorithm (GA) with Boolean Constraint Processing (BCP) is presented for solving CCOP.

The rest of this paper is organized as follows. The definition of CCOP is formalized in Section 2. The feasible solution space is reduced and transformed by Boolean Constraint Processing in Section 3. Section 4 provides an improvable GA combination of BCP. A numerical experiment is given in section 5. Section 6 provides conclusion and the future research directions.

II. PROBLEM DESCRIPTION & DEFINITIONS

CAST logic

The specification of a Bayesian Network requires an exponential number of parameters for model specification. As a model grows larger, this requirement presents a very big challenge to a system modeler. As an attempt to overcome this limitation, Chang et al. [11] developed a formalism called CAusal STrength (CAST) logic to elicit the large number of conditional probabilities from a small set of user-defined parameters. The logic has its roots in the Noisy-Or approach.

The logic requires only a pair of parameter values for each dependency relationship between any two random variables. The values are converted into conditional probability tables and the resultant tables are used during the probability propagation phase. A brief explanation of the CAST logic is provided below with the help of an example shown in Figure 1.
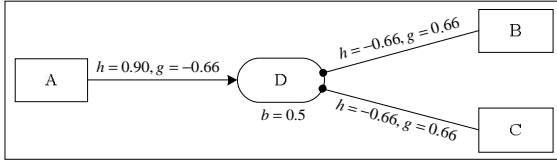
Figure 1.   An Influence Network with CAST logic parameters

Figure 1 contains three edges. On each arc, two causal strengths denoted as h and g are specified. These numbers represent the probability that a specified state of a parent node will cause a certain state in the child node. Positive values on arcs are causal influences that cause a node to occur with some probability, while negative values are influences that cause the negation of a node to occur with some probability. For instance, the arc between B and D has values $h = -0.66$ and $g = 0.66$. The first value referred to as h, states that if B is true, then this will inhibit D to be true with probability 0.66, while the second value, referred to as g, states that if B is false, then this will cause D to be true with probability 0.66. Both h and g can take values in the interval $(-1, 1)$. All non-root nodes are assigned a baseline probability, which is similar to the "leak" probability in the Noisy-Or approach. This probability is the user-assigned assessment that the event would occur independently of the modeled influences in a net. For instance, the baseline probability of D is denoted as $b = 0.5$.

There are four major steps in the CAST logic algorithm that convert the user-defined parameters into conditional probabilities. Once these steps are completed, the traditional probability calculations are performed to derive the cumulative likelihood of any event included in the Influence Net. The four steps that are performed for each conditioning case associated with the child event in the model are:

(a) Aggregate positive causal strengths.
(b) Aggregate negative causal strengths.
(c) Combine the positive and negative causal strengths.
(d) Derive conditional probabilities.

As shown in Figure 1, there are eight conditional probabilities that need to be computed to obtain the marginal probability of D. Mathematically, the marginal probability of D is computed as

$P(D) = P(D | ABC)P(A)P(B)P(C) + P(D | AB\neg C)P(A)P(B)P(\neg C)$
$+ P(D | A\neg BC)P(A)P(\neg B)P(C) + P(D | A\neg B\neg C)P(A)P(\neg B)P(\neg C)$
$+ P(D | \neg ABC)P(\neg A)P(B)P(C) + P(D | \neg AB\neg C)P(\neg A)P(B)P(\neg C)$
$+ P(D | \neg A\neg BC)P(\neg A)P(\neg B)P(C) + P(D | \neg A\neg B\neg C)P(\neg A)P(\neg B)P(\neg C)$

The four steps, described above, are used to calculate each of these eight conditional probabilities. For instance, to calculate the probability $P(D | \neg AB\neg C)$, the h value on the arc connecting B to D and the g value on the arc connecting A and C to D are considered. Hence, the set of causal strengths is denoted as C={−0.66, −0.66, 0.66}.

Step 1 Aggregate the positive causal strengths

In this step, the set of causal strengths with positive influence are combined. They are aggregated using the equation

$$PI = 1 - \prod_{c \in C \wedge c > 0} (1-c) \qquad (1)$$

Where c is the corresponding g or h value having positive influence and PI is the combined positive causal strength.

Step 2 Aggregate the negative causal strengths

In this step, the causal strengths with negative values are combined. The equation used for aggregation is

$$NI = 1 - \prod_{c \in C \wedge c < 0} (1+c) \qquad (2)$$

Where c is the corresponding g or h value having negative influence and NI is the combined negative causal strength.

Step 3 Combine causal strengths

In this step, aggregated positive and negative influences are combined to obtain an overall net influence. The difference of these aggregated influences is taken. The overall influence is obtained by taking the ratio of this difference and the corresponding promoting or inhibiting influence. Mathematically,

$$AI = \begin{cases} (PI - NI)/(1 - NI) & PI \geq NI \\ (NI - PI)/(1 - PI) & PI < NI \end{cases} \qquad (3)$$

Step 4 Derive conditional probabilities

In the final step, the overall influence is used to compute the conditional probability value of a child for the given combination of parents.

$$P(child \,|\, set\ of\ parent\ states) =$$
$$\begin{cases} baseline + (1 - baseline) \times AI & PI \geq NI \\ baseline - baseline \times AI & PI < NI \end{cases} \qquad (4)$$

The steps explained above are repeated for the remaining seven conditional probabilities in (1). It should be noted that, if the experts had sufficient time and knowledge of the influences, then the conditional probability table for each node can be used instead of g and h values. Furthermore, after estimating the conditional probability table, if some entries do not satisfy the experts, then those entries can be modified and used for computing the marginal probability.

Influence Nets

Influence Nets are Directed Acyclic Graphs (DAG) where nodes in the graph represent random variables, while the edges between pairs of variables represent causal relationships. The modeling of the causal relationships is accomplished by connecting a set of actionable events and a set of desired effects through chains of cause and effect relationships. The strength of such relationships is specified using the CAST logic parameters instead of the probabilities. The required probabilities are internally generated by the CAST logic with the help of user-defined parameters. The Influence Nets are therefore appropriate for the following situations: i) for modeling situations in which it is difficult to fully specify all conditional probability values, ii) estimates of conditional probabilities are subjective, and iii) estimates for the conditional probabilities cannot be obtained from empirical data, e.g., when modeling potential human reactions and beliefs.

The following items characterize an IN while a formal definition is given in Definition 1.

1. A set of random variables that makes up the nodes of an IN. All the variables in the IN have binary states.

2. A set of directed links that connect pairs of nodes.

3. Each link has associated with it a pair of CAST logic parameters that shows the causal strength of the link (usually denoted as h and g values).

4. Each non-root node has an associated CAST logic parameter (denoted as the baseline probability), while a prior probability is associated with each root node.

Definition 1 Influence Net

An Influence Net is a four-tuple (V, E, C, B) where

V: set of Nodes,

E: set of Edges,

C represents causal strengths: $E \rightarrow \{(h, g)$ such that $-1 < h, g < 1\}$,

B represents baseline or prior probabilities: $V \rightarrow [0, 1]$.

Figure 2 shows an example of an Influence Net. Nodes A, B and C drawn as rectangles represent the actionable events (root nodes) while nodes D and E drawn as rounded rectangles represent the desired effect (non-root nodes). The directed edge with an arrowhead between two nodes shows the parent node promoting the chances of a child node being true, while the roundhead edge shows the parent node inhibiting the chances of a child node being true. The value denoted as h and g associated with arcs shows CAST logic parameters of causal strength. The value seen within non-root nodes shows CAST logic parameters of baseline probabilities. The text associated with the non-root nodes represent the corresponding conditional probability values obtained from above CAST logic parameters while the text associated with the root nodes represents the prior probabilities.

The probability propagation in an IN is based on the "independence of parents" assumptions (similar to the loopy belief propagation [12, 13]). Thus, the marginal probability of a non-root node is computed with the help of its conditional probability table (CPT) and the prior probabilities of its parents. In this way, marginal probabilities are propagated in the forward direction, i.e., from the root nodes to the leaf nodes.
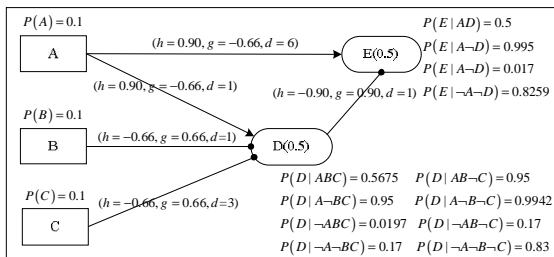

Figure 2. A Sample Influence Net

Modeling of CCOP

The object of solving CCOP is finding best course of action from a set of actions under constraints, and these actions enable to engender maximum operation effect.

The universality of CCOP containing n actions can be formalized as follow:

$$\max \quad obj(x_1, x_2, ..., x_n)$$
$$s.t. \begin{cases} a_{i1}x_1 + a_{i2}x_2 + ... + a_{in}x_n \leq b_i \\ f(x_1, x_2, ..., x_n) = 1 \\ x_j = \{0, 1\} \\ a_{ij} \geq 0, b_i \geq 0 \\ i = 1, 2, ..., m; j = 1, 2, ..., n \end{cases} \quad (5)$$

Where, Boolean variable x_j describes whether the jth action is executed. Constant b_i describes the total number of the ith kind of weapon that can be used by all the potential actions. Coefficient a_{ij} describes the quantity of the ith kind of weapon needed when executing the jth action. The Boolean function $f(x_1, x_2, ..., x_n)$ that is usually expressed as Conjunction Normal Form (CNF), describes the collaboration constraints among actions.

The actions of CCOP can be mapped to actionable nodes of IN one to one. That is, if the jth action is executed the priority probability distribution of jth actionable node is expressed as P($A_j = 1$) = 1, and P($A_j = 0$) = 0; contrarily, P($A_j = 0$) = 1, and P($A_j = 1$) = 0.

Assuming the effect node M of IN that represents the desired effect of mission, and then the object function $obj(x_1, x_2, ..., x_n)$ of CCOP can be described as the margin probability of M, where M = 1. The best solution means that choosing a set of actions that not only satisfies all the constraints but also maximizes desired effects.

The equation formula $f(x_1, x_2, ..., x_n) = 1$ is Boolean constraint, and the in-equation $a_{i1}x_1 + a_{i2}x_2 + ... + a_{in}x_n \leq b_i$ is Pseudo-Boolean constraint.

III. TRANSFORMATION & REDUCTION OF CONSTRAINTS

For pseudo-Boolean constraint $h_i(x_1, x_2, ..., x_n) \leq b_i$, where $h_i(x_1, x_2, ..., x_n) = a_{i1}x_1 + a_{i2}x_2 + ... + a_{in}x_n$, we define threshold Boole function as follow

$$g_i(x_1, x_2, ..., x_n) = \begin{cases} 1, h_i(x_1, x_2, ..., x_n) \leq b_i \\ 0, h_i(x_1, x_2, ..., x_n) > b_i \end{cases} \quad (6)$$

Apparently, if $\{x_k = \tau_k\}$ satisfies $h_i(x_1, x_2, ..., x_n) \leq b_i$, it must satisfy $g_i(x_1, x_2, ..., x_n) = 1$.

Then we define a Boolean function as follow:

$$g(x_1, x_2, ..., x_n) = \prod_{i=1}^{m} g_i(x_1, x_2, ..., x_n) \quad (7)$$

For (7), if $\{x_k = \tau_k\}$ satisfies all pseudo-Boolean constraints of CCOP, it must satisfy $g(x_1, x_2, ..., x_n) = 1$.

According to Boole's expansion theorem of literature [14], the following equation is derived:

$$g(x_1, \mathbf{K}, x_k, \mathbf{K}, x_n) = (x_k + g_{\neg x_k})(\neg x_k + g_{x_k}) \quad (8)$$

Where

$$g_{\neg x_k} = g(x_1, \mathbf{K}, 0, \mathbf{K}, x_n), g_{x_k} = g(x_1, \mathbf{K}, 1, \mathbf{K}, x_n) \quad (9)$$

Theorem 3.1 For Boolean function $g(x_1, x_2, ..., x_n)$, $\forall 1 \leq k \leq n$, $g_{\neg x_k} g_{x_k} = g_{x_k}$ is satisfied.

Proof:

If $g_{\neg x_k} g_{x_k} = 1$, apparently $g_{x_k} = 1$;

If $g_{x_k} = 0$, apparently $g_{\neg x_k} g_{x_k} = 0$;

If $g_{x_k} = 1$, then $\forall g_{r,x_k} = 1$, it is to say linear constraint $a_{r1}x_1 + \ldots + a_{rk} + \ldots + a_{rn}x_n \leq b_r$ is consistent, for $a_{rk}, b_r \in \mathbf{Z}^+$, $a_{r1}x_1 + \ldots + a_{rk-1}x_{k-1} + a_{rk+1}x_{k+1} + \ldots + a_{rn}x_n \leq b_r$ is consistent too, so $\forall g_{r,\neg x_k} = 1$, and $g_{\neg x_k} g_{x_k} = 1$.

If $g_{\neg x_k} g_{x_k} = 0$, then $g_{x_k} = 0$, proof ends;

If $g_{\neg x_k} = 0$, then $\exists g_{r,\neg x_k} = 0$, i.e. the linear constraint $a_{r1}x_1 + \ldots + a_{rk-1}x_{k-1} + a_{rk+1}x_{k+1} + \ldots + a_{rn}x_n > b_r$ is consistent, for $a_{rk}, b_r \in \mathbf{Z}^+$, $a_{r1}x_1 + \ldots + a_{rk} + \ldots + a_{rn}x_n > b_r$ is also consistent, so $g_{r,x_k} = 0$ and $g_{x_k} = 0$.

According to Theorem 3.1, it can be concluded that:

$$(x_k + g_{\neg x_k})(\neg x_k + g_{x_k})$$
$$= x_k \neg x_k + x_k g_{x_k} + \neg x_k g_{\neg x_k} + g_{\neg x_k} g_{x_k}$$
$$= \neg x_k g_{\neg x_k} + X_k g_{x_k} + g_{x_k} = \neg x_k g_{\neg x_k} + g_{x_k}(x_k + 1)$$
$$= \neg x_k g_{\neg x_k} + g_{x_k} = \neg x_k g_{\neg x_k} + g_{\neg x_k} g_{x_k}$$
$$= g_{\neg x_k}(\neg x_k + g_{x_k})$$

So (8) can be reduced to

$$g(x_1, \mathbf{K}, x_k, \mathbf{K}, x_n) = \neg x_k g_{\neg x_k} + g_{x_k} \qquad (10)$$

Apply (10) repeatedly, $g(x_1, x_2, \ldots, x_n)$ can be reduced to a common Boolean function in which only $x_1, x_2, \ldots, x_n$ included. The reduction process is as follows:

$$g(x_1, x_2, \ldots, x_n) =$$
$$\neg x_1 g(0, x_2, \ldots, x_n) + g(1, x_2, \ldots, x_n)$$
$$= \neg x_1 \neg x_2 g(0,0,x_3,\ldots,x_n) + \neg x_1 g(0,1,x_3,\ldots,x_n)$$
$$+ \neg x_2 g(1,0,x_3,\ldots,x_n) + g(1,1,x_3,\ldots,x_n)$$

$$\mathbf{K}$$

$$= \underbrace{\neg x_1 \neg x_2 \cdots \neg x_n g(0,0,\ldots,0) + \mathbf{K} + g(1,1,\ldots,1)}_{2^n}$$

There, the formula like $g(0,0,\ldots,0)$ is called discriminator, and $\neg x_1, \neg x_2, \ldots, \neg x_n g(0,0,\ldots,0)$ is called sub-item.

Apparently, for any discriminator, it can be supposed that:

$$g(0,1,0,1,0,1\underbrace{\mathbf{K},1}_{n-5})$$

Its corresponding sub-item is:

$$\neg x_1 \neg x_3 \neg x_5 g(0,1,0,1,0,1\underbrace{\mathbf{K},1}_{n-5})$$

There, $\neg x_1, \neg x_3, \neg x_5$ is the corresponding CNF of sub-item.

Apparently, if discriminator $g(\underbrace{0\mathbf{K}0}_{k},1\underbrace{\mathbf{K}}_{n-k},1) = 0$, then $g(\underbrace{0\mathbf{K}0}_{k-1},1\underbrace{\mathbf{K}}_{n-k+1},1) = 0$. If $g(\underbrace{0\mathbf{K}0}_{k_c},1\underbrace{\mathbf{K}}_{n-k_c},1) = 0$, but $g(\underbrace{0\mathbf{K}0}_{k_c+1},1\underbrace{\mathbf{K}}_{n-k_c-1},1) = 1$, then, k_c is critical value

If $k_c \geq 1$, $g(x_1, x_2, \ldots, x_n)$ can be reduced to

$$g(x_1, x_2, \ldots, x_n)$$
$$= \neg x_1 \neg x_2 \ldots \neg x_{k_c}$$
$$\underbrace{(\neg x_1 \neg x_2 \cdots \neg x_k g(0,\mathbf{K},0,\ldots,0) + \mathbf{K} + g(0,\mathbf{K},0,1,\ldots,1))}_{2^{n-k_c}}$$

Finally, the constraints of CCOP can be expressed by a common Boolean function as follow:

$$h(x_1, x_2, \cdots, x_n) = g(x_1, x_2, \cdots, x_n) f(x_1, x_2, \cdots, x_n) = 1 \quad (11)$$

Where $h(x_1, x_2, \ldots, x_n)$ only include $x_1, x_2, \ldots, x_n$.

For solving CCOP easily, $h(x_1, x_2, \ldots, x_n)$ needs to be transformed into its equivalence Disjunctive Normal Form (DNF) as follow:

$$DNF(x_1, x_2, \mathbf{K}, x_n) = \sum_{k=1}^{M} S_k(x_{k1}, x_{k2}, \mathbf{K}, x_{kl}) \quad (12)$$

Where $\{x_{k1}, x_{k2}, \ldots, x_{kl}\}$ is a subset of $\{x_1, x_2, \ldots, x_n\}$, the Boolean function $S_k(x_{k1}, x_{k2}, \ldots, x_{kl})$ is a conjunction, M is the total number of elementary conjunction.

Optimization algorithm based on DNF is designed in Section 4. The transformation from common Boolean function to DNF is mainly based on distributive law.

According to the properties of DNF, every conjunction can correspond with one feasible solution space. Let S_k corresponds with Ω_k.

For instance, assuming $S(x_1, x_2, x_3) = x_1 \wedge x_2 \wedge \neg x_3$ is a conjunction of $DNF(x_1, x_2, \ldots, x_n)$, if $x_1 = 1, x_2 = 1, x_3 = 0$, then, $S(x_1, x_2, x_3) = 1$, and thus, $DNF(x_1, x_2, \ldots, x_n) = 1$. As a result the feasible solution space depending on S is:

$$\Omega = \left\{ (1,1,0,x_4,\ldots,x_n) \,\middle|\, x_k \in \{0,1\}, 4 \leq k \leq n \right\} \quad (13)$$

Thus the feasible solution space of CCOP can be expressed as $\Omega = \bigcup_{k=1}^{M} \Omega_k$.

Apparently, $\Omega_i \cap \Omega_j$ may be not empty, thus if M is larger, then the feasible solution space is less compact.

A minimization DNF has the least conjunction, thus if $DNF(x_1, x_2, \ldots, x_n)$ is enabled to transform minimization, then the feasible solution space may be more compact.

The minimization of DNF can be captured by Quine-Mccluskey (Q-M) method [15], which is presented by algebraization of Karnaugh Map algorithm.

IV. Design of Genetic Algorithm

The variables of CCOP hold Boolean character, thus the chromosomes in GA population take the form of binary strings. Each chromosome can be thought of as a point in the search space of candidate solutions. Because of binary coding, the traditional roulette wheel selection operator, single point crossover operator and single point mutation operator are utilized by GA. The CCOP is a class of maximum problem, and the value of its object function is between 0 and 1, so the finesse function of chromosome can be defined by object function directly.

Furthermore, three new operators are introduced by improvable GA, for combination of BCP. These operators are Transformation & Reduction of Constraints, Creation of Initial Population and Feasible Solution Conversion

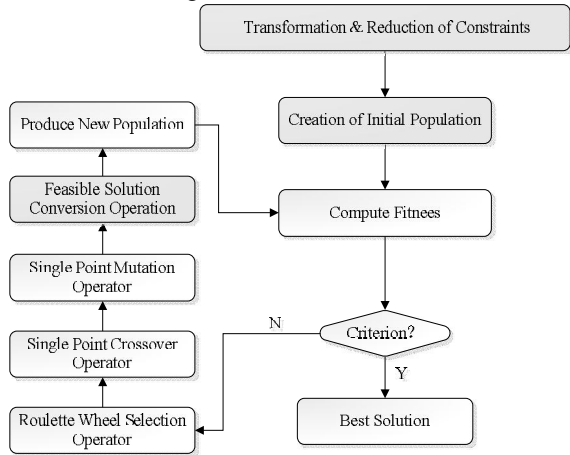Operation. The primary process of GA combination of BCP is shown in Figure 3.

Figure 3.   Process of GA combination of BCP

Transformation & Reduction of Constraints has been addressed in Section III. The next two sub-sections will explain Creation of Initial Population and Feasible Solution Conversion Operation.

Creation of Initial Population

Definition 2 Free Variable

$S_k(x_{k1}, x_{k2}, \ldots, x_{kl})$ is a conjunction of (12), the variables of set $W_k = \{x_1, x_2, \ldots, x_n\}/\{x_{k1}, x_{k2}, \ldots, x_{kl}\}$ are named as free variables, $|W_k|$ is the number of free variables.

For instance, assuming $S(x_1, x_2, x_3) = x_1 \wedge x_2 \wedge \neg x_3$ is a conjunction of (12), the number of free variables is $n{-}3$.

TABLE I.   CREATION OF INITIAL POPULATION

Algorithm 1: Creation of Initial Population
Step1: *Setting $k = 0$ and the size of population is C;*
Step2: *Computing all unitary proportion of free variables:* $$P_l = \frac{2^{|W_l|}}{2^{|W_1|} + 2^{|W_2|} + \mathbf{K} + 2^{|W_m|}}, \ 1 \le l \le M;$$
Step3: *Utilizing roulette wheel selection to select a conjunction S_l included* (12).
Step4: *Creating a chromosome randomly based on the feasible solution space Ω_l. $k = k + 1$, if $k = C$, stop, if not, go to* Step3.

In Algorithm 1, the conjunction containing more free variables has greater probability to be chosen. This can ensure the uniformity of initial population.

Feasible Solution Conversion Operation

Definition 3 Filter Constant

$S_k(x_{k1}, x_{k2}, \ldots, x_{kl})$ is a conjunction of (12), the value of variables in the set of $\{x_{k1}, x_{k2}, \ldots, x_{kl}\}$ satisfies $S_k = 1$, and the value of free variables is 0 or 1 denoted as $*$. The values according to the above rules will produce a binary string, which is defined as S_k-correlated Filter Constant.

For instance, assuming $S(x_1, x_2, x_3) = x_1 \wedge x_2 \wedge \neg x_3$ is a conjunction of (12), S-correlated filter constant is $110*\ldots*$ (the number of $*$ is $n{-}3$).

Definition 4 Filter Operation

$S_k(x_{k1}, x_{k2}, \ldots, x_{kl})$ is an item of (12), the S_k-correlated Filter Constant is $filter_k = t_1 t_2 \ldots t_n$, $list = \tau_1 \tau_2 \ldots \tau_n$ is a chromosome, then Filter Operation is defined as follow.

$$S_k \oplus ch = \sum_{i=1}^{n} t_i \otimes \tau_i \qquad (14)$$

Where, if $t_i = \tau_i$ or $t_i = *$, then $t_i \otimes \tau_i = 1$, else $t_i \otimes \tau_i = 0$.

Assuming a S-correlated Filter Constant is $110*\ldots*$, for the chromosome $ch = 1000\ldots0$, $S \oplus ch = 1$; while for the chromosome $ch = 1101\ldots1$, $S \oplus ch = 0$.

Apparently, for every chromosome ch, if there exists a conjunction S_k of (12) that satisfies $S_k \oplus ch = 0$, then, the corresponding solution of ch is a feasible solution.

Definition 4 Feasible Solution Conversion Operation

$S_k(x_{k1}, x_{k2}, \ldots, x_{kl})$ is an item of (12), $ch = \tau_1 \tau_2 \ldots \tau_n$ is a chromosome whose solution is non-feasible. Feasible Solution Conversion Operation is defined as follow:

$$ch^{,} = S_k \ominus ch \qquad (15)$$

Where, if $filter_k[i] = *$, then $ch'[i] = ch[i]$, else $ch'[i] = filter_k[i]$ ($1 \le i \le n$). $filter_k$ is a S_k-correlated filter constant.

TABLE II.   FEASIBLE SOLUTION CONVERSION OPERATION

Algorithm 2: Feasible Solution Conversion Operation
Step1: *Setting $k = 1$, current population set is pop[k], the size of population is C;*
Step2: *If $k = C$, stop; if not, choosing the chromosome pop[k];*
Step3: *Doing Filter Operation with pop[k] and every conjunction respectively, and the results are stored in array res[M];*
Step4: *If there exists an element which is 0 in res[M], goto* Step2*; if not, go to* Step5*;*
Step5: *Selecting conjunction S of the least element of res[M] randomly, doing Feasible Solution Conversion Operation pop[k] = S $\ominus$ pop[k] , then go to* Step2.

In Algorithm 2, we make the least changes of the original chromosome in the conversion from non-feasible to feasible solution. This can keep the diversity of population.

V. SIMULATION & RESULTS

This section presents an example to illustrate the feasibility and effectiveness of GA combination of Pseudo-Boolean constraint processing.

There are three kinds of weapons that used in this example, and they are denoted as weapon1, weapon2, and weapon3, respectively. The weapon requirements of every action are listed in Table 3.

TABLE III.   WEAPON REQUIREMENTS

Action	Weapon Requirements		
	Weapon1	*Weapon2*	*Weapon3*
A01	0	0	4
A02	4	8	0
A03	0	0	2
A04	6	10	0
A05	0	0	6
A06	0	0	4
A07	4	8	0
A08	6	12	0
A09	8	12	0
A10	0	0	4
A11	0	0	4
A12	0	0	2

Where the total number of weapon1 is 18, the total number of weapon2 is 36, and the total number of weapon3 is 24.

We assume that there are 12 potential actions, and the collaboration among actions is as follows:

- A01 and A02 can't be selected together;
- If A03 is selected, A06 must be selected;
- A04 and A07 can't be selected together;
- Only one of A11 and A12 can be selected;
- If A10 is selected, A08 must be selected;
- If A05 is selected, A09 must be selected.

The IN modeling of the influences of these potential actions to the mission is shown in Figure 4.
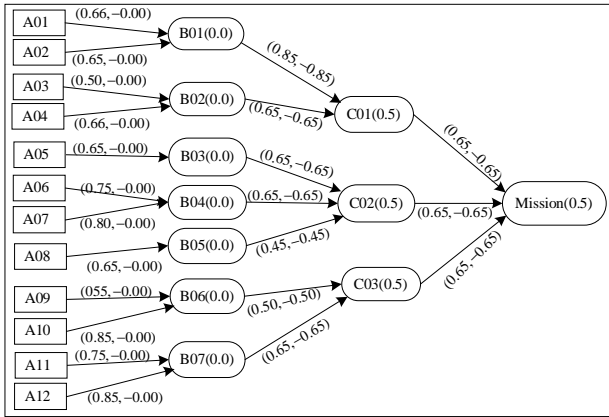

Figure 4.   The IN of actions to mission

The CCOP of this example is formalized as follow:

$$\max \quad obj(x_1, x_2, ..., x_{12})$$

$$s.t. \begin{cases} 4x_1 + 2x_3 + 6x_5 + 4x_6 + 4x_{10} + 4x_{11} + 2x_{12} \le 18 \\ 8x_2 + 10x_4 + 8x_7 + 12x_8 + 12x_9 \le 36 \\ 4x_2 + 6x_4 + 4x_7 + 6x_8 + 8x_9 \le 24 \\ (\neg x_1 \vee \neg x_2) \wedge (\neg x_3 \vee x_6) \wedge (\neg x_5 \vee x_9) \wedge (\neg x_{10} \vee x_8) \\ \wedge (\neg x_4 \vee \neg x_7) \wedge (x_{11} \vee x_{12}) \wedge (\neg x_{11} \vee \neg x_{12}) = 1 \\ x_i = \{0,1\}, i = 1, 2, ..., 12 \end{cases}$$

obj is computed by the margin probability of effect node Mission, where the value of Mission is 1.

This paper compares GA combination of Boolean constraint processing named GA_1 with GA combination of penalty function named GA_2.

The population size, maximum generation, crossover probability and mutation probability of both GAs are 100, 50, 0.90, and 0.10 respectively.

For GA_1, Transformation & Reduction of Constraints is executed firstly, and the constraints can be expressed as a DNF which consists of 80 conjunctions. Part of the conjunctions is listed in Table 4.

TABLE IV.   SOME ELEMENTARY CONJUNCTIONS

Conjunction	Filter Constant	Free Variable
$\neg x_2 \wedge \neg x_3 \wedge \neg x_4 \wedge \neg x_5 \wedge \neg x_{10} \wedge x_{11} \wedge \neg x_{12}$	*0000****010	$x_1 x_6 x_7 x_8 x_9$
$\neg x_2 \wedge \neg x_3 \wedge \neg x_4 \wedge \neg x_5 \wedge x_8 \wedge x_{11} \wedge \neg x_{12}$	*0000**1**10	$x_1 x_6 x_7 x_9 x_{10}$
$\neg x_2 \wedge \neg x_4 \wedge \neg x_5 \wedge x_6 \wedge x_8 \wedge x_{11} \wedge \neg x_{12}$	*0*001*1**10	$x_1 x_3 x_7 x_9 x_{10}$
…	…	…
$\neg x_1 \wedge \neg x_3 \wedge \neg x_4 \wedge \neg x_7 \wedge x_8 \wedge x_9 \wedge \neg x_{11} \wedge x_{12}$	0*00**011*01	$x_2 x_5 x_6 x_{10}$

For GA_2, the Boolean constraint can be transformed into numerical constraint as follow refer to [10].

$$(\neg x_1 \vee \neg x_2) \wedge (\neg x_3 \vee x_6) \wedge (\neg x_5 \vee x_9) \wedge (\neg x_{10} \vee x_8) \wedge (\neg x_4 \vee \neg x_7) \wedge$$
$$(x_{11} \vee x_{12}) \wedge (\neg x_{11} \vee \neg x_{12}) = 1 \Leftrightarrow (2 - x_1 - x_2)(1 - x_3 + x_6)(1 - x_5 + x_9)$$
$$(x_{11} + x_{12})(1 - x_{10} + x_8)(2 - x_4 - x_7)(2 - x_{11} - x_{12}) \ge 1$$

The penalty function of GA_2 is defined as follow.

$$f_1(x_1, x_2, \mathbf{K}, x_{12}) = 4x_1 + 2x_3 + 6x_5 + 4x_6 + 4x_{10} + 4x_{11} + 2x_{12} - 24$$
$$f_2(x_1, x_2, \mathbf{K}, x_{12}) = 8x_2 + 10x_4 + 8x_7 + 12x_8 + 12x_9 - 36$$
$$f_3(x_1, x_2, \mathbf{K}, x_{12}) = 4x_2 + 6x_4 + 4x_7 + 6x_8 + 8x_9 - 18$$
$$f_4(x_1, x_2, \mathbf{K}, x_{12}) = 1 - (2 - x_1 - x_2)(1 - x_3 + x_6)(1 - x_5 + x_9)$$
$$(x_{11} + x_{12})(1 - x_{10} + x_8)(2 - x_4 - x_7)(2 - x_{11} - x_{12})$$

$$fun(x_1, x_2, ..., x_{12}) = \exp(-M \times \sum_{i=1}^{4} \max(f_i(x_1, x_2, ..., x_{12}), 0))$$

Then, the fitness function of GA_2 is defined as follow.

$$fit = obj(x_1, x_2, ..., x_{12}) \times fun(x_1, x_2, ..., x_{12}) \quad (16)$$

The result of GA_1 is 101011111110, that is, the selected actions are A01   A03   A05   A06   A07   A08   A09   A10 and A11, and the *obj* is 0. 7555. The result of GA_2 is 10110101011, that is, the selected actions are A01   A03   A04   A06   A08   A11 and A12, and the *obj* is 0. 7487. The process of simulation is shown in Figure 5.
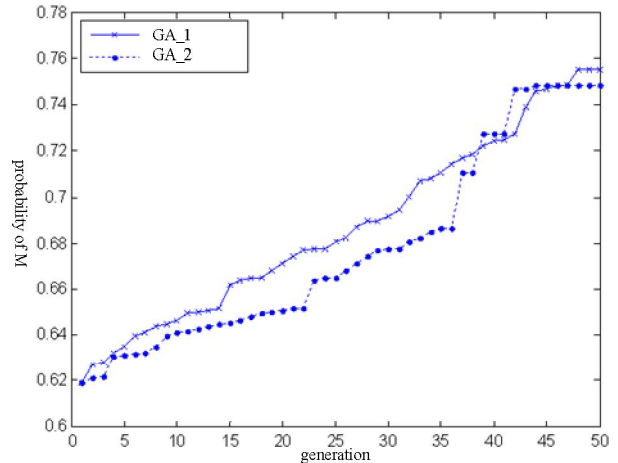

Figure 5.   Process of simulation

As shown in Figure 5, the optimization result of GA_1 is better than GA_2.

The chromosomes contained within population of GA_1 are all feasible solutions, while some chromosomes are generally non-feasible solutions in every population of GA_2. Thus, the search effectiveness of GA_1 is better than GA_2 potentially.

On the other hand, the penalty function of GA_2 easily reduces the diversity of population, which may result in the phenomenon of premature convergence. While BCP injected GA enables population to keep diversity effectively. Thus the convergence velocity of GA_1 is slower than GA_2. Further, the optimal solution found by GA_1 is more probable.

VI. CONCLUSIONS

This paper overcomes the limitation of Influence Nets based course of action optimization that lacking in fully considering the constraints among actions, and presents a new class of CCOP in Influence Net by introducing resource and collaboration constraints. Then, a novel method of GA combination of BCP is presented to solve CCOP. The feasibility of improvable GA has been proved by mathematics, and then this paper presents an example to illustrate the effectiveness of the GA above. The research has been applied in the domain of military.

REFERENCES

[1] Rosen, J. A. and Smith, W. L., "Influence Net Modeling with Causal Strengths: An Evolutionary Approach," Proceedings of the 1996 Command and Control Research and Technology Symposium, Monterey CA, June 1996, pp. 699–708.

[2] Erez Karpas, Solomon Eyal Shimony and Amos Beimel, "Approximate belief updating in max-2-connected Bayes networks is NP-hard," Artificial Intelligence, vol.173 (12-13), August 2009, pp.1150–1153.

[3] Lee W. Wagenhals and Larry K. Wentz, "New Effects-Based Operations Models in War Games," Proceedings of the 8th International Command and Control Research and Technology Symposium, Washington DC, June 2003.

[4] Sajjad Haider, Abbas K. Zaidi and Alexander H. Levis, "A Heuristic Approach for Best Set of Actions Determination in Influence Nets," In Proceedings of IEEE International Conference on Information Reuse and Integration, Las Vegas, November 2004, pp. 600–605.

[5] Sajjad Haider, Abbas K. Zaidi and Alexander H. Levis, "Identification of Best Set of Actions in Influence Nets," International Journal of Hybrid Intelligent Systems (IJHIS), vol.5 (1), January 2008, pp. 19–29.

[6] M.M. Ali, Z. Kajee-Bagdadi, "A local exploration-based differential evolution algorithm for constrained global optimization," Applied Mathematics and Computation, vol.208 (1), February 2009, pp.31–48.

[7] WANG Yong, CAI Zi-Xing, ZHOU Yu-Ren, XIAO Chi-Xin, "Constrained Optimization Evolutionary Algorithms," Journal of Software, vol. 20(1) ,January 2009, pp. 11–29.

[8] Helio J. C. Barbosa, Afonso C. C. Lemonge. "An Adaptive Penalty Method for Genetic Algorithms in Constrained Optimization Problems", Frontiers in Evolutionary Robotics, InTech, April 2008, pp.9–34.

[9] Özgür Yeniay. "Penalty function methods for constrained optimization with genetic algorithms". Mathematical and Computational Applications, vol.10 (1), January 2005, pp. 45–56.

[10] JI Xiao-Hui, HUANG Zhuo, ZHANG Jian. "On the Integration of Constraint Programming and Optimization," Chinese Journal of Computers, vol.28 (11), November 2005, pp. 1790–1797.

[11] K.C. Chang, P.E. Lehner, A.H. Levis, S.A.K. Zaidi, X. Zhao, "On causal influence logic," technical report, George Mason University, Center of Excellence for C3I, December 1994.

[12] F.R. Kschischang, B.J. Frey, "Iterative decoding of compound codes by probability propagation in graphical models," IEEE Journal on Selected Areas in Communication, vol.16, February 1998, pp. 219–230.

[13] McEliece, R. J., MacKay, D. J. and Cheng, J. F., "Turbo Decoding as an Instance of Pearl's Belief Propagation Algorithm," IEEE Journal on Selected Areas in Communication, vol.16, February 1998, pp. 140–151.

[14] Frank Markham Brown, "Boolean Reasoning," Dover Publications, January 2003.

[15] Tarun Kumar Jain, D. S. Kushwaha and A. K. Misra, "Optimization of the Quine-McCluskey Method for the Minimization of the Boolean Expressions," Proceedings of 4th International Conference on Autonomic and Autonomous Systems, Washington DC, November 2008, pp.165-168.

Yanguang Zhu received the Master of Engineering in System Engineering from the National University of Defense Technology, Changsha, China, in 2006. He is a full-time PhD student in Military Equipment of the National University of Defense Technology at present.

During his master study, he helped to develop an optimization framework for missile general design with multi-method collaboration support. Currently his research interest covers simulation and assessment for Weapon Equipment System of Systems. He has published more than 20 papers. 10 were indexed by EI.

Dongliang Qin received the Bachelor of Engineering in System Engineering from the National University of Defense Technology, Changsha, China, in 2008. He is a full-time PhD student in Military Equipment of the National University of Defense Technology at present.

During his bachelor study, he helped to develop software for course of action assessment. Currently his research interest covers modeling and simulation of System of Systems.

Yifan Zhu received the Bachelor of Science in Mechanics from the Peking University, Beijing, China, in 1983, received the Master of Engineering in Solid Mechanics from the National University of Defense Technology, Changsha, China, in 1989, received the Doctor of Engineering in System Engineering from the National University of Defense Technology, Changsha, China, in 2003

His main research currently interest covers assessment and optimization for emergency plan based on causality modeling, that is supported by the National Natural Science Foundation of China under grants No 91024015. He has published three research books, two tutorial book, and 60 papers.

Xingping Cao received the Bachelor of Science in Mathematics, Master and Doctor of Engineering in Control Science and Engineering from the National University of Defense Technology, Changsha, China, in 1997, 2000 and 2004, respectively.

She is currently lecturer of the National University of Defense Technology. Her main research interests include simulation experiment management, simulation evaluation. She has published more than 20 papers.