

Research Article

An Innovative Thinking-Based Intelligent Information Fusion Algorithm

Huimin Lu,^{1,2} Liang Hu,¹ Gang Liu,² and Jin Zhou¹

¹ School of Computer Science and Technology, Jilin University, Changchun 130012, China

² School of Software, Changchun University of Technology, Changchun 130012, China

Correspondence should be addressed to Huimin Lu; luhm.cc@gmail.com

Received 8 April 2013; Accepted 30 April 2013

Academic Editors: L. He and P. Melin

Copyright © 2013 Huimin Lu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This study proposes an intelligent algorithm that can realize information fusion in reference to the relative research achievements in brain cognitive theory and innovative computation. This algorithm treats knowledge as core and information fusion as a knowledge-based innovative thinking process. Furthermore, the five key parts of this algorithm including information sense and perception, memory storage, divergent thinking, convergent thinking, and evaluation system are simulated and modeled. This algorithm fully develops innovative thinking skills of knowledge in information fusion and is a try to converse the abstract conception of brain cognitive science to specific and operable research routes and strategies. Furthermore, the influences of each parameter of this algorithm on algorithm performance are analyzed and compared with those of classical intelligent algorithms through test. Test results suggest that the algorithm proposed in this study can obtain the optimum problem solution by less target evaluation times, improve optimization effectiveness, and achieve the effective fusion of information.

1. Introduction

Information fusion [1] refers to the process in which relevant information is searched and extracted from multiple distributed heterogeneous network resources and then converted into a unified knowledge mode. It aims at constructing effective knowledge resources for solving the problems in certain field or generating new integrative knowledge object by conversing, integrating, combining, and so forth various information coming from distributed information resources.

Common information fusion algorithms can be divided into two main categories, which are probability statistics method and artificial intelligence method. Probability statistics method includes Bayes, the transformation of Bays [2], and D-S evidence reasoning [3]. It has axiomatic basis and low computational complexity and is intuitive and easy to be understood, but it needs more prior information and its applicable condition is harsher; while in artificial intelligence method, information fusion is similarly regarded as that human brain comprehensively treats information. In this method, artificial neural network [4], support vector machine [5], and genetic algorithm (GA) [6] account for

approximately 85% of the whole information fusion algorithm. And the machine learning methods, that is, swarm intelligence, artificial immune, quantum genetic algorithm, and so forth, have been applied in information fusion. This method shows fewer requirements to the prior information of object and stronger self-fitness. Moreover, the fusion of the subjective and objective information in system can be realized using this method. Most of existing intelligent algorithms are proposed based on natural evolution rule, animal collective intelligence, and life system mechanism. However, they fail to make good use of the background factors of problems and the knowledge produced in the process of solving problems. This situation limits the natural combination of intelligent algorithm and knowledge to some extent and the full play of the role of knowledge.

The researches on information fusion currently show a developing trend of further combining with the cognitive system-based human natural intelligence. The cognitive behaviors of humans, such as learning and reasoning, handle various information by widely simulating human intelligence from each link of information carrier to fusion based on knowledge [7]. Innovative thinking is the advanced stage of

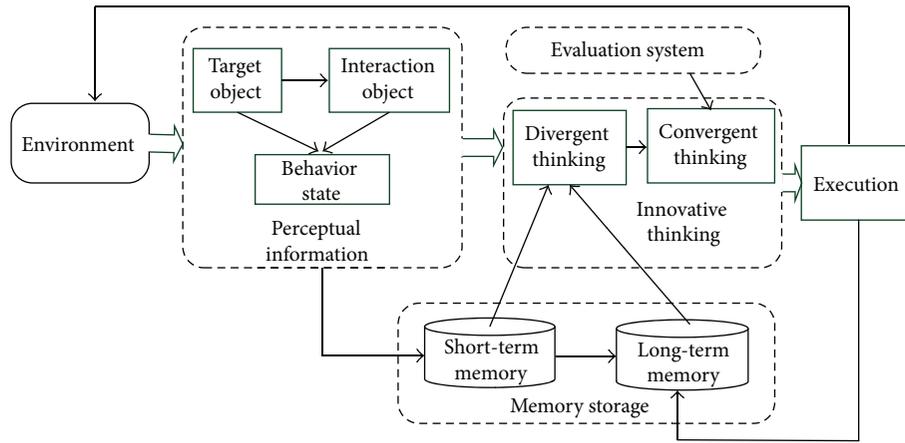


FIGURE 1: The basic framework and the relationship of the five parts of ITIIF algorithm.

thinking activities. It is a thinking process of discovering new things, creating new methods, and solving new problems on the basis of personal experiences. Moreover, it is the source of wisdom [8, 9]. Through modeling and simulating innovative thinking by computer, the innovative ability on human level can be obtained and produced. By simulating human thinking mechanism, it is expected to reach a higher solving efficiency than that obtained by the natural selection mechanism-based intelligent algorithm [10].

Therefore, in references to the relevant research achievements of brain cognitive theory and creativity computation, this study puts forward an intelligent algorithm which can realize information fusion—innovative thinking-based intelligent information fusion algorithm (ITIIF algorithm) by the algorithmization to cognitive process and behavior of innovative thinking.

2. The Basic Framework of ITIIF Algorithm

ITIIF algorithm treats knowledge as core and information fusion as a knowledge-based innovative thinking process. It gives full play to the role of the knowledge-based innovative thinking skills in information fusion. This algorithm is composed by five parts, which are information sense and perception, memory storage, innovative thinking, evaluation system, and execution. Among which, innovative thinking includes two thinking skills, which are divergent thinking and convergent thinking. The basic framework and the relationship of the five parts of this algorithm are shown in Figure 1.

ITIIF algorithm can be expressed by sextuple as

$$ITIIF = \langle P, M, D, C, V, E \rangle, \tag{1}$$

where P is the perceptual information module, M is memory storage module, D is divergent thinking module, C is convergent thinking module, V is evaluation system module, and E is execution module. The functions of each module are as follows.

- (1) Perceptual information module: it reasons and fuses the information collected from environment and

maps this information into knowledge. It includes information sense and information perception.

- (2) Memory storage module: it organizes and manages knowledge; it includes long-term memory (LTM) and short-term memory (STM).
- (3) Divergent thinking module: it integrates the physical world information world obtained and information world knowledge into a structural knowledge divergent model. This process can be regarded as a self-organizing process of the knowledge in human brain. It assembles the knowledge fragments into a knowledge mode with certain structure (tree or figure).
- (4) Convergent thinking module: it transforms the knowledge divergent model produced by divergent thinking into decision by focus, modification, or innovation.
- (5) Evaluation system module: it is the assessment standard for the decision satisfaction degree. It is used to evaluate the whole or part of decision plan. In this module, value system will affect thinking modules.
- (6) Execution module: it applies the decision produced by thinking into environment to realize intelligent control.

3. The Modules of ITIIF Algorithm

3.1. Information Sense and Information Perception. (1) Information sense: it means that certain attributive character or behavior states of the target object collected are processed according to rule and historical data to acquire accurate values. It is represented as

$$F\text{-sense } (A, B, C, \dots) = S1(a, b, c, \dots). \tag{2}$$

$F\text{-sense } (A, B, C, \dots)$ denotes the processing of information sense, where A , B , and C are the sets of the multiple values of certain sense information collected and a , b , and c are the accurate values obtained by sense information processing.

```

(1) Set  $RO = \emptyset$  ( $RO$  is related object set);  $R = \emptyset$  ( $R$  is relationship set);  $S1 = \emptyset$  ( $S1$  is sense information set);
(2) while ( $i < FS.length$ ) ( $FS$  is the set of various sense information) do
(3) Process sense information:  $new\_sense_i \leftarrow F\text{-sense}(fs_i)$ ;
(4) Add  $new\_sense_i$  to  $S1$ ;
(5) end
(6) Recognize objects:  $RO \leftarrow \text{Pattern\_rec}(S1)$  ( $\text{Pattern\_rec}()$  is object reorganization);
(7)  $tar\_o \leftarrow tar1$  ( $tar1$  is target object);
(8) while ( $i < RO.length$ ) do
(9) if ( $r_i = \text{Relate}(tar\_o, RO_i) > \Delta_{re}$ , ( $\text{Relate}()$  is relevancy calculation,  $\Delta_{re}$  is threshold);
(10)  $HashSet_0 \leftarrow ro_i, HashSet_1 \leftarrow r_i$ 
(11) end
(12)  $KC = \{tar\_o, HashSet_0, HashSet_1\}$ 
    
```

ALGORITHM 1: KC_construction algorithm.

For example, the temperature information in a room is obtained by multiple temperature sensors distributed in different positions $T = (21^\circ, 22^\circ, 37^\circ, 24^\circ, 23^\circ)$. Firstly, based on rules, there is $t \in [t_0 - \delta, t_0 + \delta]$, where t is room temperature, t_0 is the historical data measured, and δ is temperature range control parameter. Assume that $t_0 = 22^\circ$, $\delta = 5^\circ$, 37° will be deleted as an outlier. Then, based on averaging rule, the current room temperature is calculated as $t = (21^\circ + 22^\circ + 24^\circ + 23^\circ)/4 = 22.5^\circ$.

(2) Information perception: it maps a plurality of sense information into knowledge cluster by reasoning and integrating. It is expressed as

$$F\text{-perception: } S1(a, b, c, \dots) \rightarrow KC, \quad (3)$$

where F-perception: $S1(a, b, c, \dots)$ is the processing of cognitive information and KC refers to knowledge cluster.

Definition 1 (knowledge cluster (KC)). It refers to the knowledge structure connected by target object (including attribute and behavior) and related objects; it is expressed as

$$KC = \{tar_o, (ro_1, ro_2, \dots, ro_i), (r_1, r_2, \dots, r_i) \mid ro_i \in O, r_i \in Re\}, \quad (4)$$

where tar_o represents target object, ro_i is related object, O is object set, and Re is the set of the relationships of target object and related objects.

For example, in a traffic intersection, the KC formed with an automobile as object is shown in Figure 2.

Thus the knowledge cluster in Figure 2 can be expressed as {automobile, (traffic police, pedestrian A, pedestrian B, traffic lane, vehicle 2, vehicle 1, traffic light), (0.92, 0.5, 0.46, 0.6, 0.8, 0.7, 0.9)} (Algorithm 1).

3.2. Memory Storage. Memory storage is divided into LTM and STM; it is denoted as

$$M = \langle LTM, STM \rangle \quad (5)$$

STM is used to store the knowledge cluster constructed by information perception as well as thinking process data. LTM is used to store the knowledge unit produced by innovative thinking. It can carry out the automatic evolution of knowledge.

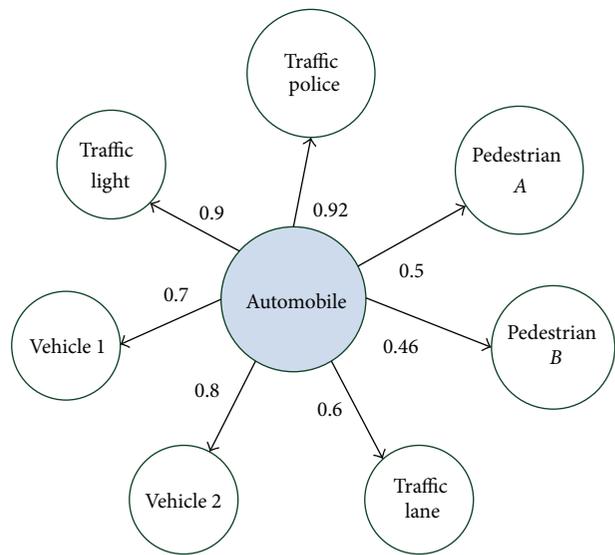


FIGURE 2: The knowledge cluster formed with an automobile as object.

3.2.1. STM. STM is a temporary memory space opened in thinking process for the development of thinking activities. STM can be expressed as

$$STM = \{kc_i \mid i = 1, 2, \dots, n\}, \quad (6)$$

where kc_i represents an independent knowledge cluster, $kc_i = \langle D_{kc_i}, t_{kc_i} \rangle$, D_{kc_i} denotes the stimulation intensity of KC, which suggests the influencing degree of KC on target environment. Bigger D_{kc_i} explains that KC cannot be easily forgotten; t_{kc_i} is the existing time of KC in STM. The greater the t_{kc_i} is, the more easily the knowledge cluster can be forgotten.

3.2.2. LTM. LTM is the container of knowledge units. The knowledge units obtained by innovative thinking are stored in LTM.

Definition 2 (knowledge unit). It refers to the knowledge structure connected by subobjects in sequence to achieve the expected state of system. It can be expressed as

$$KU = \{(sg_i, sg_j, r_{ij}) \mid sg_i, sg_j \in \Omega, r_{ij} \in SG_{Re}\}, \quad (7)$$

where Ω is subtarget space, SG_{Re} denotes the set of the interaffecting relationship of sg_i and sg_j to realize the expected effect of system.

LTM is expressed as

$$LTM = \{ku_i \mid i = 1, 2, \dots, n\}. \quad (8)$$

ku_i is an independent knowledge unit, $ku_i = \langle F_{ku_i}, t_{ku_i} \rangle$, F_{ku_i} is the fitness of knowledge unit. Larger F_{ku_i} indicates a stronger activeness of knowledge unit, and this knowledge unit can more easily be extracted by thinking module. t_{ku_i} refers to the time that knowledge unit is stored in LTM. Larger t_{ku_i} suggests that the knowledge unit will be more easily forgotten.

3.2.3. Knowledge Evolution. The knowledge units in LTM continuously evolve under the effect of innovative thinking activities. This evolution contains quality development and quantity growth. The quantity growth of knowledge represents the increase of total knowledge quantity in LTM after a certain period. The quality development of knowledge refers to the improvement of the depth and truth degree of the recently emerged knowledge units produced by innovative thinking comparing with that in certain past historical period. By quality development of knowledge, the knowledge units in LTM can be updated and continuously evolve.

The specific steps of knowledge evolution are as follows.

- (1) Initialize knowledge evolution scale Pk .
- (2) $list(ku_1, ku_2, ku_3, \dots, ku_n) // list()$ sorts the knowledge units in LTM from high to low by fitness, $ku_{n-1}.fitness() > ku_n.fitness()$, where $fitness(ku_j) = w_1p_{j1} + w_2p_{j2} + \dots + w_n p_{jn}$, $p_{j1}, p_{j2}, \dots, p_{jn}$ is the evaluation value of the $1, 2, \dots, n$ subtarget in the j knowledge unit, and w_n is weight.
- (3) $select(ku_1, ku_2, ku_3, \dots, ku_{pk}) // select()$ select the first Pk knowledge units into evolution pool.
- (4) The knowledge units in evolution pool are stochastically paired (ku_i, ku_j) , and $i > j, i = 1, 2, \dots, pk, j = 1, 2, \dots, pk$.
- (5) By the recombination and local mutation to the attribute gene corresponded to the chromosome of knowledge unit, new knowledge unit is generated $\{ku_{new1}, ku_{new2}, \dots, ku_{newn}\}$.
- (6) Test the effectiveness of new knowledge units: if $ku_{newn} \cdot effectiveness() > Th$, this new knowledge unit is stored in LTM, where Th is threshold. $effectiveness(ku_{newn}) = w_1A(ku_{newn}) + w_2B(ku_{newn}) + w_3C(ku_{newn})$, where $A(ku_{newn})$ is correctness of knowledge unit, $B(ku_{newn})$ is the coverage degree of knowledge unit, and $C(ku_{newn})$ is the reliability of knowledge unit.

$A(ku_{newn})$ measures the degree of knowledge unit ku_{newn} realizing the expected target of system, $A(ku_{newn}) = est()$. $est()$ is target estimation function. Its estimation accuracy is related to the precision of knowledge and estimation method.

$B(ku_{newn})$ represents the ratio of the number of subtarget contained in ku_{newn} and the number of sub-target contained in realizing the expected target. $B(ku_{newn}) = \sum_{i=1}^{|SG_{ku_{newn}}|} (sg_i \in SG_{ku_{newn}} \cap SG_{\Omega}) / |SG_{\Omega}|$, where $SG_{ku_{newn}}$ is the sub-target contained in knowledge unit ku_{newn} , $|SG_{ku_{newn}}|$ is the cardinal number of $SG_{ku_{newn}}$, and SG_{Ω} denotes the sub-target contained in realizing the expected target of system in theory.

$C(ku_{newn})$ is the correctness degree of knowledge unit, $C(ku_{newn}) = (1 + ((T_{\Omega} - 1) / T_{LTM})) / T_{\Omega}$; in this formula, T_{Ω} indicates the number of knowledge unit in realizing the expected target of system in theory; T_{LTM} is the number of knowledge unit in realizing expected target of system contained in LTM.

3.3. Divergent Thinking. Divergent thinking is a process of assembling the sub-targets in different knowledge units to produce new target path according to the characteristics of problem and the process required in assembling knowledge units [11]. It extracts and assembles certain knowledge units in memory storage to achieve the reflection from LTM to knowledge tree. It is represented as

$$\text{divergent: LTM} \longrightarrow \text{KST}, \quad (9)$$

where KST refers to the knowledge structure tree assembled; divergent composes two processes, which are memory extraction and knowledge assembling.

3.3.1. Memory Extraction. The extraction of knowledge units is controlled by the fitness F_{ku_i} of the knowledge units and existing time t_{ku_i} . And the memory extraction model is established as

$$GK(ku_i) = F_{ku_i} e^{-\lambda t_{ku_i}}, \quad (10)$$

where $GK(ku_i)$ is the extraction intensity of the i knowledge unit. Larger $GK(ku_i)$ suggests that knowledge unit can more possibly be extracted at divergent thinking stage. Divergent thinking will extract C_{LTM} knowledge units with larger $GK(ku_i)$ value from LTM. C_{LTM} is the extraction capacity of knowledge unit; it determines the number of the maximum knowledge units that can be extracted from LTM.

3.3.2. Knowledge Assembling. Knowledge assembling combines the knowledge units extracted by memory into knowledge structure tree (KST). Several definitions that are related to KST are shown in the following.

Definition 3 (knowledge structure tree). It refers to the data structure produced by combining the knowledge units in certain order according to the characteristics of problem. It is presented as

$$\text{KST} = \langle g, SG, DB, P \rangle, \quad (11)$$

where g is target node; it reflects the purposiveness of divergent thinking. There is only one target node in KST; SG is sub-target node set; it refers to all subtargets that are related to problem solution; DB is directed border set; P is the relationship parameter maintained by each node.

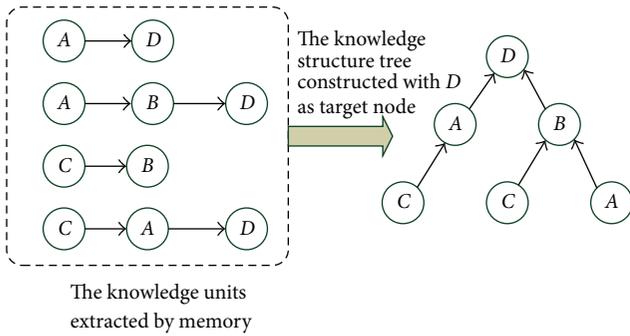


FIGURE 3: The assembling process of knowledge structure tree.

Definition 4 (knowledge path). If there is a node sequence $\{node_m, node_1, node_2, \dots, node_k, node_n\}$ in KST and correlations $parentOf(node_m, node_1)$, $parentOf(node_1, node_2)$, and $parentOf(node_k, node_n)$ belong to KST, there is a knowledge path $P(node_m, node_n)$ between $node_m$ and $node_n$. $parentOf(node_i, node_j)$ refers to that $node_i$ is the parent node of $node_j$.

Definition 5 (knowledge radius). If there is a knowledge path $P(node_m, node_n)$ between $node_m$ and $node_n$ in KST, the node conversion times on this path are called the distance between $node_m$ and $node_n$. And the shortest distance is the knowledge radius of $node_m$ and $node_n$.

Figure 3 presents the assembling process from knowledge unit to KST.

Each knowledge path from source node to target node represents one problem solving plan. Source node is the sub-target node containing no subnodes. Divergent thinking is, namely, a process of listing all problem solving plans according to existing knowledge level and cognitive conditions. Every knowledge path can be regarded as a special kind of knowledge unit. But there is no one-to-one corresponding relationship between knowledge path and knowledge unit. All knowledge units can find their corresponding knowledge paths in KST, but not all knowledge paths in KST can find their corresponding knowledge units. This phenomenon suggests that divergent thinking has combining and innovating capacity. For example, the knowledge unit extracted by the knowledge path $C \rightarrow B \rightarrow D$ in KST from memory does not exist, it is generated by the combination of knowledge units $C \rightarrow B$ and $A \rightarrow B \rightarrow D$. That is, the combination of $C \rightarrow B$ and $A \rightarrow B \rightarrow D$ creates a problem solution plan $C \rightarrow B \rightarrow D$.

Knowledge assembling constructs KST based on the method of knowledge radius and with knowledge target as center. The specific KST construction algorithm is as follows (see Algorithm 2).

3.4. Convergent Thinking. Multiple problem solving plans have been produced through assembling and innovating knowledge units by divergent thinking, while there is a focus process when human's thinking proceeds to certain stage. After focusing, usually several limited objects are retained.

This is the process of convergent thinking. Convergent thinking refers to the process that the divergent knowledge model formed by divergent thinking is transformed into new problem solving plans by focusing, modifying, and innovating. It realizes the reflection from KST to the optimum problem solving plan. It is indicated as

$$\text{convergent: KST} \rightarrow \text{CS.} \tag{12}$$

3.4.1. Convergent Operation Focus(). Focus() can select several paths with higher values form multiple knowledge paths of KST. Its convergent rule is

$$F = \langle f(kp_i), n_{con} \rangle, \tag{13}$$

where $f(kp_i)$ is focus function, $kp_i \in \text{KST}$ represents all knowledge paths included in KST, n_{con} is convergent dimension, namely, the number of the knowledge paths focused.

Focus function $f(kp_i) : \text{KST} \rightarrow [0, 1]$ maps all the knowledge paths in KST into $p \in [0, 1]$. p reflects the focused probability of each knowledge path. The knowledge path with larger p suggests that the probability of this knowledge path being in focus is higher. The knowledge path with higher values in KST poses higher convergent probability. Thus $f(kp_i) = V_{kp_i}(\text{KST})$; $V_{kp_i}(\text{KST})$ is the values of the i knowledge path in KST.

3.4.2. Modified Operation: Modified(). When the problem solving value achieved by the focused knowledge path shows little differences with expected target value, modified() is operated. It is represented as

$$\text{modified}(sg_i, G, E, R), \tag{14}$$

where sg_i is the sub-target modified; G is the expected target achieved by implementing modified operation; E is the environment of modified operation, namely, the affecting relationship of G and the integrity degree of the sub-targets except sg_i in focused knowledge path. R is resource restriction. It refers to the usable resources in the premise of ensuring the resource demand of other sub-targets.

3.4.3. Innovation Operation: Innovate(). When the problem solving value achieved by the focused knowledge path shows a larger difference with expected target value, innovate() is conducted. It is expressed as

$$sg_{new} = \text{innovate}(FKP_{sg_i}), \tag{15}$$

where $sg_{new} \in \Omega$ is the new target after innovation. FKP_{sg_i} is the i sub-target on focused knowledge path.

Aiming at different problems and different problem solving stages, human thinking will choose different innovation styles (modified operation or innovation operation) according to environment changes:

$$\text{MI}(kp, \text{cre_style}, v_e, \alpha, \beta), \tag{16}$$

```

(1) Set  $r = 1$  ( $r$  is the variable of knowledge radius) and  $set\_T_{000} \leftarrow g$  ( $g$  is goal)
(2) while ( $r \leq R$ ) do
(3)   for ( $i = 0; i < |set\_T_{ri}|; i++$ ) do
(4)     for ( $j = 0; j < |set\_T_{rij}|; j++$ ) do
(5)       for  $\forall ku \in KU_{LTM}$  ( $ku$  is knowledge unit) do
(6)         for  $\forall sg \in ku$  ( $sg$  is the sub-target of knowledge unit  $ku$ ) do
(7)           if  $g$  is a parent node  $sg$  then
(8)              $set\_T_{rij} \leftarrow sg$ 
(9)           end
(10)        end
(11)     end
(12)  end
(13)   $r = r + 1;$ 
(14) end

```

ALGORITHM 2: KST_construction algorithm.

where kp is the knowledge path produced by focus operation, $cre_style \in \{0, 1\}$ is creative style. $cre_style = 1$ refers to dynamic creative style parameter; $cre_style = 0$ is static creative style parameter; v_e is target expected value; $\alpha \in [0, 1]$ is modified operation coefficient; $\beta = 1 - \alpha$ is innovation operation coefficient. If $cre_style = 1$, the style is dynamic creative style; its parameter is defined as

$$\alpha^* = 1 - \frac{v_e - v}{v_e}, \quad \beta^* = 1 - \alpha^*, \quad (17)$$

where v refers to the value that can be achieved by current knowledge path. It can be known that, with the increase of v , the probability of implementing modified innovation is gradually enhanced, while the probability of implementing innovative innovation is gradually weakened.

3.5. Evaluation System. Evaluation system aims at evaluating the satisfaction degree to problem solving plan, which is, evaluating the values of the knowledge paths contained in KST. The satisfaction degree of the i knowledge path is denoted as

$$SAT(KST_{kp_i}) = 1 - \frac{V_{ge} - V(KST_{kp_i})}{V_{ge}}, \quad (18)$$

where V_{ge} is expected values of target; it is decided by decision makers; $V(KST_{kp_i})$ is the values of the i knowledge path in KST. The specific calculation formula is as follows:

$$V(KST_{kp_i}) = w_1 V(kp_i(sg_1)) + w_2 V(kp_i(sg_2)) + \dots + w_j V(kp_i(sg_j)). \quad (19)$$

w_j is weight; $V(kp_i(sg_j))$ is the contribution value provided by the j sub-target in the i knowledge path in achieving the total target; $V(kp_i(sg_j)) = f(t_{sg_j}, l_{sg_j}, n_{sg_j})$, $f(t_{sg_j}, l_{sg_j}, n_{sg_j})$ are the evaluation functions related to the memory time, the layer number in KST, and the subnodes number of sub-target j . $V(kp_i(sg_j))$ is closely connected with knowledge expression form. Therefore, the calculation mode of $f(t_{sg_j}, l_{sg_j}, n_{sg_j})$ should be instantiated when facing specific problem.

4. ITIIF Algorithm Realization

According to the analysis on each module of ITIIF algorithm, the specific realization of ITIIF algorithm is explained in Algorithm 3.

5. Experiment and Analysis

5.1. Experiment and Analysis on the Parameters of ITIIF Algorithm. The experimental data employ the data in ‘‘The Famous Knowledge Base for OpenCyc’’. By sorting, test data set is formed. It includes 37,310 knowledge units and 6, 531 correlations between knowledge units. This test aims at analyzing the parameters (C_{LTM} , n_{con} , cre_style) in ITIIF algorithm using knowledge navigation path optimization problem. First, we define two indexes—thinking set degree and information quantity.

Definition 6 (thinking set degree). It refers to the aggregation degree of the solutions obtained by convergent thinking. It is expressed as

$$SE(t) = \sum_i \frac{k_i(t)}{t} \ln \frac{k_i(t)}{t}, \quad (20)$$

where t is evolution generation number, $SE(t)$ is the thinking set degree of the t generation. $k_i(t)$ is the total focused times of the i solution that has been focused up to the t generation.

$IQ(t)$ refers to the information quantity obtained after evolution of t generations. It is expressed as

$$IQ(t) = \frac{n_f(t)}{2n^2}, \quad (21)$$

where $n_f(t)$ refers to the different knowledge unit number in current LTM. n is the knowledge unit number in corpus.

The influences of parameter C_{LTM} , n_{con} , cre_style on the performance of ITIIF algorithm are shown in Figures 4, 5, and 6, respectively.

As shown in Figure 4, C_{LTM} shows great influences on the optimization effect of navigation path. In view of the whole

- (1) Parameter initialization: memory extraction capacity C_{LTM} , convergent dimension n_{con} , creative style cre_style
- (2) Generate sense information set FS and conduct sense information treatment to $FS: F -sense(fs_i) \Rightarrow S1$
- (3) Perform mode reorganization and KS construction to $S1: KC_construct(S1) \Rightarrow STM$.
- (4) Construct knowledge units by learning: $ku_i = learn(TSM) \Rightarrow LTM$
- (5) Proceed divergent thinking to the knowledge units in LTM ($divergent(LTM)$) according to specific problem, and assemble KST
- (6) Evaluate all the knowledge paths in $KST(SAT(KST_{kpi}))$ and operate convergent thinking;
 - (i) Conduct focus operation ($focus(KST_{kpi})$) to obtain the optimum problem solving plan.
 - (ii) if $cre_style = 0$, conduct static innovation; if $cre_style = 1$, conduct dynamic innovation. Dynamic innovation mode is:
 - (a) if $\alpha^* \geq Th$ (Th is threshold), conduct modified operation: $modified(sg_i, G, E, R) \Rightarrow ku_{new}$
 - (b) if $\alpha^* < Th$ proceed innovation operation: $innovate(FKP_{sg_i}) \Rightarrow ku_{new}$;
- (7) Execute decision plan $execute(ku_{new})$ and realize $ku_{new} \Rightarrow LTM$
- (8) Conduct knowledge evolution $evolute(LTM_{ku})$ in LTM.

ALGORITHM 3: ITIIF algorithm.

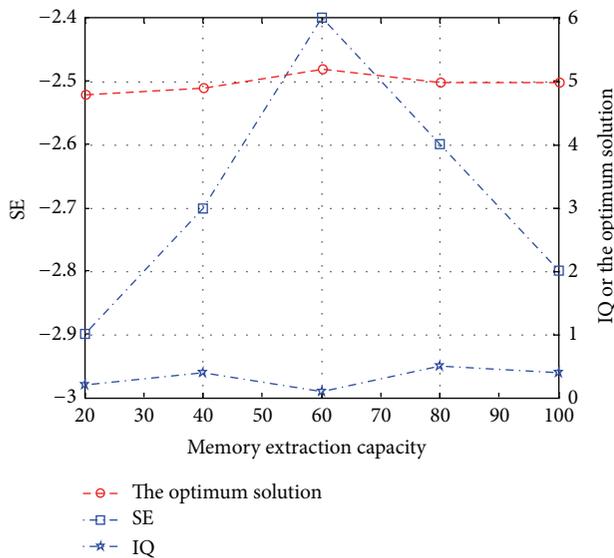


FIGURE 4: The relationships of C_{LTM} with thinking set degree, information quantity, and the optimum solution.

process, problem solving effect experiences a process that first increases then decreases with the increase of C_{LTM} ; on one hand, with the increase of C_{LTM} , thinking set degree $SE(t)$ first increases and then decreases. $SE(t)$ increase indicates that convergent thinking is easy to “focus” on certain solution with a relatively small number; on the other hand, with the increase of C_{LTM} , information quantity $IQ(t)$ firstly increases then decreases and increases again. Information quantity increase suggests that the information required in solving the problem is perceived more completely and deeply. Meanwhile, there is a law that the target value of optimum solution is proportional to $SE(t) * IQ(t)$.

In Figure 5, with the increase of convergence dimension n_{con} , optimization effect of problem solving first reaches a certain value and then decreases. This phenomenon suggests

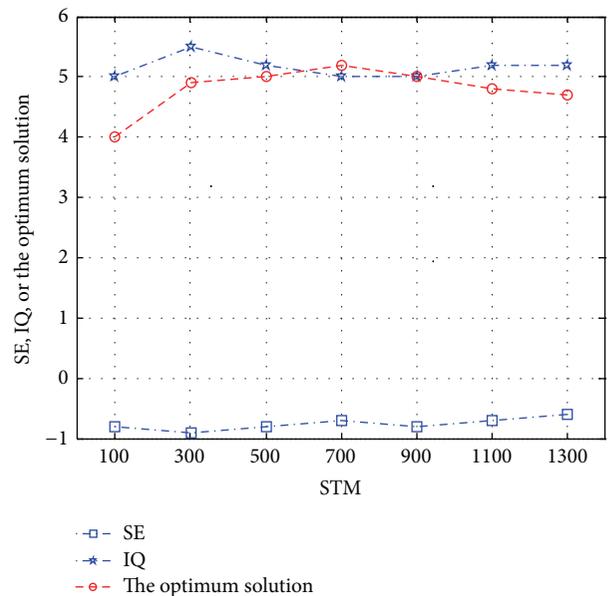


FIGURE 5: The relationships of STM with thinking set degree, information quantity, and the optimum solution.

that, with the increase of information quantity, the divergent degree of thinking will increase. But when the divergent degree is too high, problem solving effect will be influenced.

As shown in Figure 6, as for dynamic creative style $cre_style = 1$, better optimization effect can be obtained when the v_e selected is close to the optimum value. When the v_e selected is deviated from the optimum value, the performance of ITIIF algorithm reduces greatly; while as for the static creative style $cre_style = 0$, the optimizing capacity of ITIIF algorithm is continuously improved with the increase of β . In general, the $SE(t) * IQ(t)$ value obtained when dynamic creative style parameter is applied is lower than that when static creative style parameter is applied. So it can be concluded that if the probable range of the optimum value of target can be predicted, it is proper to select dynamic creative

TABLE 1: The test functions.

Test functions	Domain	Characteristics
$f_1(x) = \sum_{i=1}^n x_i^2$	$x_i \in [-5.12, 5.12],$ $i = 1, 2, \dots, n$	Unimodal function, the optimum value locates on point 0
$f_2(x) = 100(x_1^2 - x_2)^2 + (1 - x_1)^2$	$x_i \in [-2.048, 2.048],$ $i = 1, 2, \dots, n$	Unimodal function is a nonconvex and pathological function. The global optimum value is located in a smooth, long, and narrow parabolic canyon
$f_3(x) = 0.5 + (\sin^2 \sqrt{x_1^2 + x_2^2} - 0.5) / [1.0 + 0.001(x_1^2 + x_2^2)]^2$	$x_i \in [-100, 100],$ $i = 1, 2, \dots, n$	Multimodal function. It shows a ring of local minimum values on the uplift in a range that is about 3.14 away from the global minimum value. It has strong oscillation properties. And its global minimum value is surrounded by local minimum values.
$f_4(x) = (1/4000) \sum_{i=1}^n (x_i - 100)^2 - \prod_{i=1}^n \cos((x_i - 100) / \sqrt{i}) + 1$	$x_i \in [-100, 100],$ $i = 1, 2, \dots, n$	Multimodal function, the global minimum value 0 is obtained on $x_i = 0$
$f_5(x) = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10)$	$x_i \in [-5.12, 5.12],$ $i = 1, 2, \dots, n$	Ultra multimodal function, it contains about n^{11} local minimum values

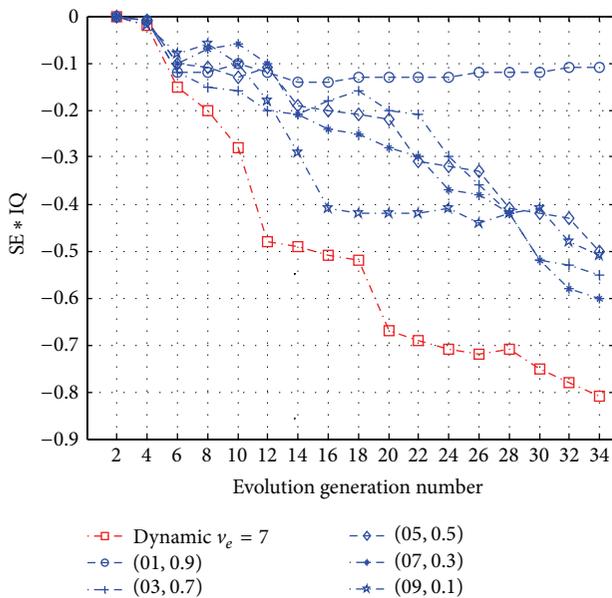


FIGURE 6: The relationship between creative style with $SE(t) * IQ(t)$.

style; if the range of target value cannot be confirmed, static creative style is more proper. At the same time, the global optimization capacity of ITIIF algorithm can be improved by increasing β value. Commonly, β is set in 0.6 to 0.9.

5.2. Comparison Test

5.2.1. Comparison of Optimization Results. To better display the performance of ITIIF algorithm, this paper adopts five testing functions to develop simulation tests. Moreover, the simulation results of ITIIF algorithm are compared with those of the currently used binary particle swarm and binary differential evolution algorithm. Table 1 lists the test functions.

To provide comparability, the evaluation indexes in this test experiment mean best fitness (MBF) and standard deviation

(SD). MBF reflects the accuracy that algorithm can achieve when iteration times are given. SD reflects the stability and robustness of algorithm. Table 2 shows the solution results of the testing functions by ITIIF algorithm, binary particle swarm algorithm, and binary differential evolution algorithm. The values of MBF and SD are obtained by independently running each algorithm for 20 times, respectively.

It can be seen from Table 2 that the ITIIF algorithm proposed in this study shows better optimization performances to the functions, regardless of unimodal function or multimodal function. Moreover, it achieves more ideal optimization effects in terms of solution accuracy and stability. This result shows that ITIIF algorithm has certain advantages in function optimization.

5.2.2. Performance Comparison. The ITIIF algorithm proposed in this study is compared with the commonly used genetic algorithms (GA), estimate distribution algorithms (EDAs), and ant colony optimization (ACO) in three aspects, which are average distance, average time, and average assessment value. The parameters settings of each algorithm are shown in Table 3.

The comparison results are presented in Figure 7. It can be obtained that ITIIF and EDAs show better optimization effects, while GA present the poorest optimization effect. The average time of ITIIF is longer than that of the other three algorithms. This is because the evolution of ITIIF algorithm requires learning and innovative thinking. And the longer average time of ITIIF algorithm is just the result of fully used knowledge. With the accumulation of knowledge, the time consumed in solving problem will be decreasing, just as human can improve the problem solving efficiency through continuous accumulations of knowledge and experience by learning. The assessment value of ITIIF is only 4.5% to 8% of other algorithms. This result means that the time required in target evaluation is greatly reduced.

According to above experimental results, it can be concluded that ITIIF algorithm shows better efficiency than

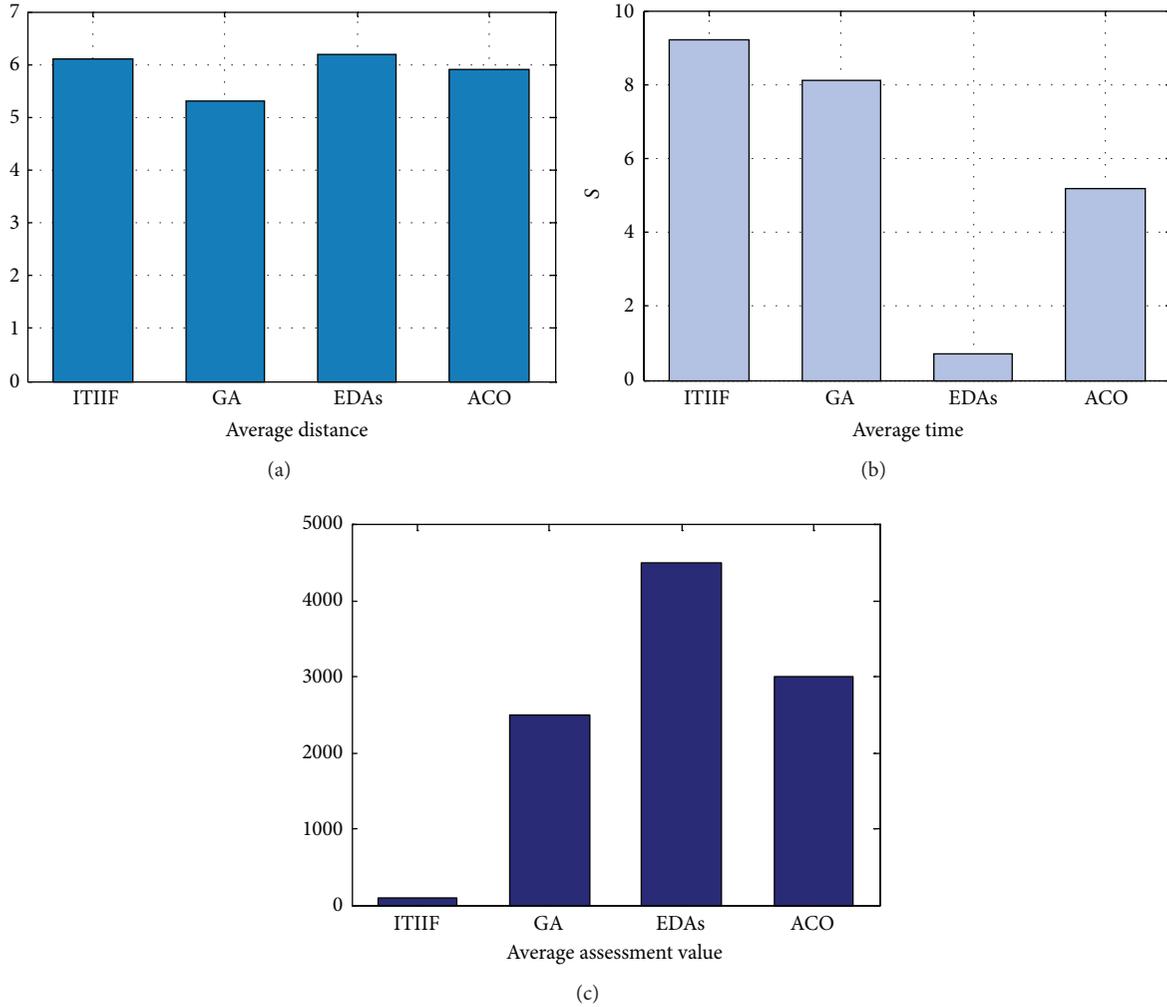


FIGURE 7: The performance comparison of ITIIF, GA, EDAs, and ACO.

TABLE 2: The solution results of the testing functions by ITIIF algorithm, binary particle swarm algorithm, and binary differential evolution algorithm.

Function	Binary particle swarm algorithm		Binary differential evolution algorithm		ITIIF algorithm proposed in this study	
	MBF	SD	MBF	SD	MBF	SD
$f_1(x)$	0.000006	0.000007	0.0	0.0	0.0	0.0
$f_2(x)$	0.000256	0.000275	0.0	0.0	0.0	0.0
$f_3(x)$	0.000001	0.000003	0.0	0.0	0.0	0.0
$f_4(x)$	37.033964	4.886305	0.561765	0.160205	0.51602917	0.2710531
$f_5(x)$	277.696409	10.621365	58.452170	10.675424	0.001025	0.000765

some of current intelligent algorithms. This is because ITIIF algorithm replaces evaluation operation with creative thinking process. Thus ITIIF algorithm can make better use of the knowledge generated by target evaluation to further improve efficiency. In addition, the evolution process of ITIIF algorithm can be explained. And ITIIF algorithm directly simulates the cognitive behavior of human and is consistent with the cognitive habits of human. This is what not possessed by other intelligent algorithms.

6. Conclusions

In references to the research achievements of modern brain science, neurophysiology, and cognitive theories, this study proposes a knowledge-based and value evaluation-oriented information fusion algorithm starting from the viewpoints of biological control theory and macro control system theory. Besides, this algorithm integrates the skills of convergent thinking and divergent thinking and applies the structure

TABLE 3: The parameters settings of each algorithm.

Algorithm	Parameter setting
ITIFF	$C_{LTM} = 50$, $C_{STM} = 10000$, $n_{con} = 1$, $cre_style = 1$
GA	Populations = 80, iterations number = 50, crossover probability = 0.9, mutation probability = 0.1
EDAs	Populations = 300, iterations number = 50, selective probability = 0.2
ACO	Information heuristic factor = 0.8, expected heuristic factor = 0.6, pheromone evaporation rate = 0.5

analysis methodology of macro system control theory into the simulation and modeling of innovative thinking. The test results show that this algorithm can reduce the number of target evaluation times, improve optimization efficiency, and realize the effective fusion of information. The following work is to strengthen strict theoretical derivation, incorporate innovative thinking into the frame of this algorithm, extensively integrate this algorithm with various traditional evolutionary algorithms, and widen the application area.

Acknowledgments

This research is sponsored by the 2011 China Postdoctoral Science Foundation Funded Project under Grant no. 2011M500611, the 2012 China National Natural Science Foundation Funded Project under Grant no. 61240029, and the 2011 Industrial Technology Research and Development Special Project of Jilin Province under Grant no. 2011006-9. This work is also supported by the Science and Technology Key Project of Jilin Province under Grant no. 2011ZDGG007, the Basic Scientific Research Project of Jilin University under Grant no. 93k172011k06.

References

- [1] H. Ruser and F. P. Leon, "Information fusion-an overview," *Technisches Messen*, vol. 74, no. 3, pp. 93–102, 2007.
- [2] S. Messaoudi, K. Messaoudi, and S. Dagtas, "Bayesian data fusion for smart environments with heterogenous sensors," *Journal of Computing Sciences in Colleges*, vol. 25, no. 5, pp. 140–146, 2010.
- [3] J. Yang, H. Z. Huang, Q. Miao, and R. Sun, "A novel information fusion method based on Dempster-Shafer evidence theory for conflict resolution," *Intelligent Data Analysis*, vol. 15, no. 3, pp. 399–411, 2011.
- [4] Z. Hong-bin, "Multi-sensor information fusion method based on the neural network algorithm," in *Proceedings of the 5th International Conference on Natural Computation (ICNC '09)*, pp. 534–536, August 2009.
- [5] Y. Wang, C. Zhang, and J. Luo, "Study on information fusion algorithm and application based on improved SVM," in *Proceedings of the 13th International IEEE Conference on Intelligent Transportation Systems (ITSC '10)*, pp. 1271–1276, Funchal, Portugal, September 2010.
- [6] Z. Wanga, K. S. Leung, and J. Wang, "A genetic algorithm for determining nonadditive set functions in information fusion," *Fuzzy Sets and Systems*, vol. 102, no. 3, pp. 463–469, 1999.
- [7] L. W. Barsalou, "Introduction to 30th anniversary perspectives on cognitive science: past, present, and future," *Topics in Cognitive Science*, vol. 2, no. 3, pp. 322–327, 2010.
- [8] G. Wen, Y. Ding, and Q. Zheng, "Overview on computational creativity," *Pattern Recognition and Artificial Intelligence*, vol. 16, no. 2, pp. 185–191, 2003.
- [9] M. A. Boden, *The Creative Mind: Myths and Mechanisms* edition, Routledge, London, UK, 2nd edition, 2004.
- [10] P. P. Bonissone, R. Subbu, N. Eklund, and T. R. Kiehl, "Evolutionary algorithms + domain knowledge = real-world evolutionary computation," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 3, pp. 256–280, 2006.
- [11] L. Gabora, "Cognitive mechanisms underlying the creative process," in *Proceedings of the 4th Creativity and Cognition Conference*, pp. 126–133, ACM, Loughborough, UK, October 2002.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

