



A recommender mechanism based on case-based reasoning

Chen-Shu Wang^a, Heng-Li Yang^{b,*}

^a Graduate Institute of Information and Logistics Management, National Taipei University of Technology, 1, Sec. 3, Chung-hsiao E. Road, Taipei, Taiwan

^b Department of MIS, National ChengChi University, 64, Sec. 2, Chihnan Road, Taipei, Taiwan

ARTICLE INFO

Keywords:

Recommender mechanism
Case-based reasoning
Multiple stage reasoning
Genetic algorithm
Artificial intelligence application

ABSTRACT

Case-based reasoning (CBR) algorithm is particularly suitable for solving ill-defined and unstructured decision-making problems in many different areas. The traditional CBR algorithm, however, is inappropriate to deal with complicated problems and therefore needs to be further revised. This study thus proposes a next-generation CBR (GCBR) model and algorithm. GCBR presents as a new problem-solving paradigm that is a case-based recommender mechanism for assisting decision making. GCBR can resolve decision-making problems by using hierarchical criteria architecture (HCA) problem representation which involves multiple decision objectives on each level of hierarchical, multiple-level decision criteria, thereby enables decision makers to identify problems more precisely. Additionally, the proposed GCBR can also provide decision makers with series of cases in support of these multiple decision-making stages. GCBR furthermore employs a genetic algorithm in its implementation in order to reduce the effort involved in case evaluation. This study found experimentally that using GCBR for making travel-planning recommendations involved approximately 80% effort than traditional CBR, and therefore concluded that GCBR should be the next generation of case-based reasoning algorithms and can be applied to actual case-based recommender mechanism implementation.

© 2011 Elsevier Ltd. All rights reserved.

1. Introduction

Case based reasoning (CBR) is a paradigm, concept, and intuitive mechanism for solving ill-defined and unstructured problems (Belecheanu, Pawar, Barson, Bredehorst, & Weber, 2003). Similar to the natural human problem-solving process, CBR retrieves past experiences for reuse in regard to target problems. Since such process is likely to need to revise previous-case solutions before applying them, CBR then retains successful problem-solving experiences for further reuse (Aamodt & Plaza, 1994). This, then, is traditional CBR's 4R processes of retrieve, reuse, revise, and retain.

CBR is therefore a classical artificial intelligence algorithm. Many have applied CBR within various problem-solving domains (Aamodt & Plaza, 1994; Kolodner, 1993; Shiu & Pal, 2001; Waston, 1997). Cirovic and Cekic (2002) applied CBR to construction projects during their preliminary design phase by retrieving historical cases from a historical project database, storing useful case(s) in their construction knowledge base, and then applying the most similar previous case(s) to improve the quality of construction designs. Belecheanu et al. (2003) referred to past records in order to reduce information uncertainty in regard to such industrial requirements as those involved in new product development, particularly when employing the concurrent engineering approach.

Furthermore, Chang (2005) applied CBR to screening children with delayed development in order to detect their disorder early through analysis of their symptoms, thereby improving the chances of effective treatment. Both Garrell, Golobardes, Bernado, and Llorca (1999) and Golobardes, Llorca, Salamo, and Marti (2002) have used CBR to diagnose breast cancer based on mammary biopsy data and micro calcifications, respectively. Additionally, Shimazu, Shibata, and Nihei (2001) applied conversational case-based algorithm (CCBR) to developing a mentor guide for user helpdesk implementation and Shimazu (2002) applied CCBR to automatic-clerk mechanisms and electronic website shopping assistance. Researchers have historically tended to solve these problems by using such mathematical models as regressions but these mathematical models involve too many assumptions to be applied effectively to real-world problem solving, and CBR seems to be a feasible alternative.

Researchers have until recently extended CBR applications as mechanisms for making recommendations based on previous cases. Yang and Wang (2009a) applied the CBR algorithm to information-system project management as a recommender mechanism by offering project managers preferences from previous cases to help project managers construct new project plans. They also applied similar mechanisms to travel-schedule planning. Educators, furthermore, can integrate CBR recommender mechanism into e-learning systems to provide learners with reference-certification paths (2009b). Such real-world problems as these are usually difficult to formulate within strict mathematical

* Corresponding author.

E-mail addresses: wangcs@ntut.edu.tw (C.-S. Wang), yanh@nccu.edu.tw (H.-L. Yang).

models, and people have often solved them using experiences they obtain by word-of-mouth. Some studies (Adomavicius & Kwon, 2007; Adomavicius & Tuzhilin, 2005) have also recommended that the next generation of recommender mechanisms should focus on real-life problem solving and applications. Case-based recommender mechanisms are therefore particularly appropriate for solving unstructured problems because people can use the CBR style to describe them and should therefore be regarded as a new problem-solving paradigm.

In order to create such a mechanism it is necessary to review, redefine, and expand both the traditional recommender mechanisms and the original CBR algorithms. Using the traditional CBR algorithm for complex problems requires retrieving each case for the decision makers' multiple objectives. As decision-making problems become increasingly complicated, however, a merely multiple-objective problem representation becomes too unsophisticated to reflect their reality. A revised case-based recommender mechanism equipped with the ability to address more complicated real-life problems is therefore necessary, as obtaining actionable information is particularly valuable for decision makers. Cao and Zhang (2007) found that existing recommender mechanisms cannot provide decision makers with a direction in which to take action, even though recommender mechanisms should be able to tell decision makers what to do next (Yang, 2007). Based on the previous cases that CBRs have retrieved, a next-generation recommender mechanism needs to have the ability to provide decision makers with better directions in regard to what actions to take.

Furthermore, traditional CBR mechanisms have to evaluate all the cases in the case base to retrieve those most similar case(s) which makes their efficiency strongly and negatively related to the size of the applicable case base. Consequently, researchers have therefore developed numerous approaches to decreasing the effort involved in case evaluation, with K-means being the most popular approaches. K-CBR, which involves integrating CBR with the k-means approach, first clusters all the cases and only evaluates those from the most similar cluster for case retrieval. Chang and Lai (2005) then attempted to apply self-organizing maps (SOMs), and found that SOM-CBR outperformed k-CBR, although both k-CBR and SOM-CBR improved CBR's efficiency. The performances of the two revised CBR mechanisms are, however, closely related to the case representation and indexing approach (Shin & Han, 1999), so their superior performances are unstable and cannot be guaranteed.

This study therefore proposes a revised case-based recommender mechanism, to which it refers as the next-generation CBR (GCBR) algorithm. GCBR is also applicable to various real-world applications, particularly case-based recommender mechanisms, and can serve as a new problem-solving paradigm. GCBR is designed to improve traditional CBR's efficiency and stability regardless of the case representation and indexing approach employed. Section 2 of this paper presents a new method for describing problems. Section 3 presents the proposed GCBR model. Section 4 reports an experiment using this model and also presents a scenario illustrating GCBR application. Section 5 presents conclusions and proposes future research directions.

2. Problem description: hierarchical criteria architecture (HCA)

Descriptions of decision-making problems involving multiple objectives become too complicated to represent them adequately (Coello, 2000) but if decision makers are unable to conceptualize such problems clearly they are unlikely to devise trustworthy and useful solutions. This study has therefore adopted a new representation methodology for describing decision-making problems called the hierarchical criteria architecture (HCA) in order to

enable decision makers to state their problems adequately (Yang & Wang, 2008). This follows Adomavicius and Tuzhilin's (2005) recommendation that a next-generation recommender system should be able to solve multi-dimensional problems.

HCA can enhance descriptions of problems involving multiple objectives by enabling decision makers to describe each decision's objectives with the appropriate amount of detail. The solutions to problems described using HCA are therefore more valuable than those using other methods because decision makers can represent such problems accurately. Fig. 1 illustrates how describing problems using HCA allows decision makers to consider them from a multi-criteria perspective while still reducing each criterion hierarchically until reaching the required level of detail with the description remaining sufficiently detailed to represent the problem. HCA is therefore an improved and enhanced methodology for presenting decision-making problems.

Fig. 2 illustrates a hypothetical e-learning system problem in which learners must use the recommender system to retrieve a similar previous case or cases for an information technology (IT) certification examination reference and learning-path suggestion. By representing this decision-making problem using HCA, learners can set three objectives for their decisions comparing the similarity of such data in the case-base as those in regard to personal demographics, capabilities, and learning paths. They may also decide that they can measure personal capabilities with work experience and thereby achieve IT certification. They can therefore increase the detail of the target conditions until they consider the problem description to be sufficiently complete.

3. The proposed GCBR model

To address the HCA problem, this study has revised Yang and Wang's (2008) revised CBR algorithm and used it to propose a case-based recommender mechanism that is a GCBR algorithm. Table 1 shows its variables and their definitions and descriptions.

The revised GCBR algorithm has three characteristics. One indicates that GCBR is a generalized problem-solving model because of its ability to help solve HCA problems. Similar to traditional CBR problems, HCA problems can include multiple objectives on a single level. Furthermore, each decision's objectives can be divided into multiple hierarchical levels. Another characteristic is that GCBR acts as a predictor in support of multi-stage decision making. GCBR can provide decision makers with a series of cases stage by

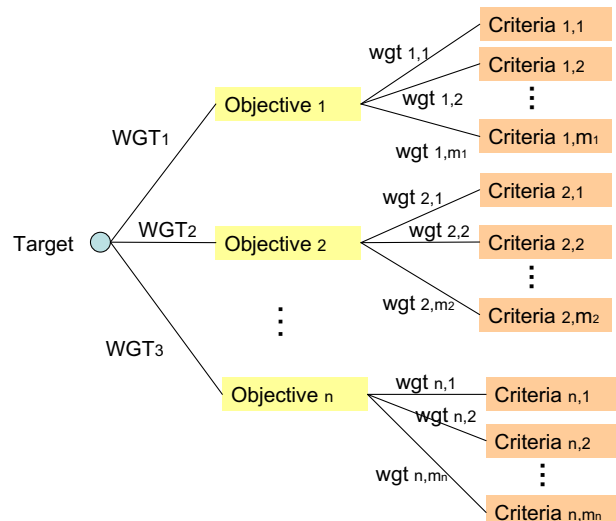


Fig. 1. HCA problem representation methodology.

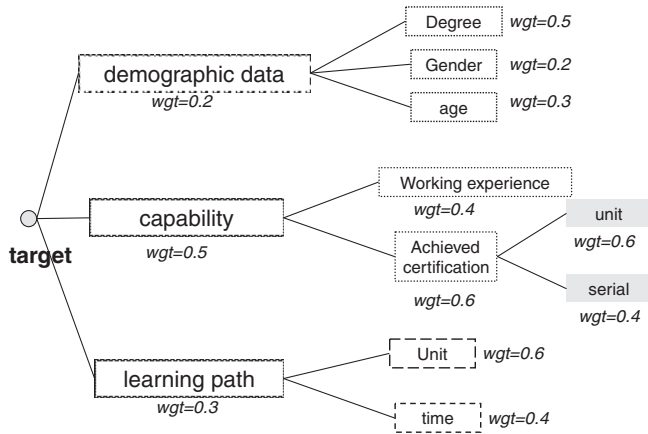


Fig. 2. An IT certification recommender problem described by HCA.

Table 1
GCBR variables definitions and description.

Variable	Definition and description
n	The number of cases in the case base
C_i	The i th case of the case base, $i = 1, 2, \dots, n$
fet	Features used to describe a case
m	The number of features each case employs
T	The target case inputted by the decision makers. The recommender mechanism provides them with feasible reference cases according to the target case's condition
fet_j	To present the j th feature, that $\begin{cases} fet_j^{C_i} & \text{the } j\text{th feature of case } i \\ fet_j^T & \text{the } j\text{th feature of target} \end{cases}$ $i = 1, 2, \dots, n; j = 1, 2, \dots, m$
wgt_{fet}	Decision makers can assign an importance weighting to the feature, wgt_{fet_j} , to denote the importance weight of the j th feature; $j = 1, 2, \dots, m$
$difference(C_i)$	$difference(C_i)$ is an array that records the degree of difference between the i th case and the target case (T), which is evaluated by Eq. (5), $i = 1, 2, \dots, n$
$gap(fet_j^{C_i}, fet_j^T)$	To evaluate the j th feature gap of the i th case and T , $i = 1, 2, \dots, n; j = 1, 2, \dots, m$, by Eq. (4)
$level(fet_j)$	To return to the condition that decision makers set on the j th feature
$threshold$	Decision makers can set the gap threshold. If the difference in a case is less than the threshold, it provides the case as a reference
$next_level(fet_j)$	To retrieve the next level feature of the j th feature
$fet_check_function$	To evaluate the gap between the j th feature and T
k	The numbers of stages, k , that decision makers expect the recommendation system to require for providing a feasible suggestion

stage suggesting reference cases in each stage. This refined information is useful for decision makers in revising their solutions. The third characteristic is that its performance is superior to that of the traditional CBR algorithm because it employs a genetic algorithm (GA) to keep the convergence rate stable, thereby increasing the efficiency of the solution process.

3.1. GCBR as a generalized problem-solving model

The traditional CBR algorithm's 4R steps are that CBR *retrieves* the feasible cases so that decision makers may either *reuse* the solution of these retrieved cases directly or *revise* the solution according to real applications; CBR then *retains* the successful case or cases and the solution in the case base for further reference. This process is similar to that of ordinary human problem solving, and many have applied CBR successfully to a variety of contexts during the past few decades.

Its core algorithm evaluates the similarity of the target case (T) with the cases in the case base. The retrieval sub-algorithm evaluates the similarity between the target and each case in the case base by summarizing each feature's gap, which describes that case in detail. CBR judges the similarity here by calculating the difference between each case and the target, the similarity increasing as the difference decreases. Each feature has a *fet-check-function* to evaluate the features' similarity, as illustrated in Fig. 3.

The *fet-check-function* is set to either total or partial similar according to the feature's characteristics. For partial similarity, the check function returns a real number between 0, indicating that they are identical, and 1, indicating that they are totally different. The total similar *fet-check-function*, however, returns either 1 or 0. For example, if the target's gender feature is female and case _{i} is male, the gap between the target and case _{i} would be 1. The difference between case _{i} and the target is therefore the gap as calculated using Eq. (1). GCBR then selects the case with the smallest gap as the most feasible solution and provides it to the decision makers for reference.

$$difference(\vec{C}_i, \vec{T}) = cosine(\vec{fet}_m^{C_i}, \vec{fet}_m^T) = \frac{\vec{fet}_m^{C_i} \cdot \vec{fet}_m^T}{\|\vec{fet}_m^{C_i}\|_2 \times \|\vec{fet}_m^T\|_2} \quad (1)$$

The case-retrieval sub-algorithm needs to be revised, however, because the problems that GCBR addresses involve hierarchical levels of criteria. This study therefore proposes the recursive sub-algorithm *fet-check-rewrite*, as shown in Fig. 3. This algorithm is a recursive one for rewriting *fet-check-function* in order to allow GCBR to manage the HCA problem. When the HCA feature level exceeds 1, such as in $level(fet) > 1$, the weighted sum of the next level replaces the *fet-check-rewrite*.

For example, according to the IT certification example illustrated in Fig. 2 the similarity evaluation function should be altered, as shown in Fig. 4, to consider three levels recursively, so its consideration of level 3, which is the dash-block area, precedes that of level 2, which precedes that of level 1. The *Fet-check-rewrite* mechanism is a function that returns the revised *feature-check-function* to GCBR's core algorithm in order to evaluate the case's similarity with the target.

Fig. 5 compares the gap between the features of specific cases with the target using the *fet-check-function*, as shown in Eq. (2). Eq. (3) then summarizes the feature gap to evaluate the similarity between the target and each case in the case base. Fig. 6 presents Fig. 5's *Reuse* algorithm. Following Yang and Wang's (2009a) procedure, GCBR then analyzes the retrieved case or cases further using the knowledge discovery (KDD) mechanism, which includes association mining techniques and statistical analyses to produce potential knowledge rules and then provide decision makers with revised case information upon which they can take action. Yang and Wang (2008) claimed that simply presenting the retrieved case or cases to decision makers is useless because the case filtering performs poorly under loose target conditions. The system should therefore employ data mining analysis to identify

```

Function fet_check_function = fet-check-rewrite (fet)
if ( level(fet) > 1)

fet_check_function = sum wgt_fet * fet_check_rewrite(next_level(fet))
else
fet_check_function = wgt_fet * fet_check_function
end if;
end
    
```

Fig. 3. Fet-check-rewrite algorithm.

knowledge with the potential to assist in decision making. Except for the retrieved cases, KDD results can also provide decision makers with refined information for revising actions in order to improve the quality of their decisions.

3.2. GCBR as a prophet recommender: multiple stages of recommendations

A next-generation recommender should also support multiple stages of recommendations. People are likely to face multi-stage decision-making problems in such situations as making travel plans for several days, in which they need a recommender mechanism that provides a suggestion package with detailed action recommendations for each stage. Almost all current case-based recommender mechanisms, however, involve only single-stage reasoning.

Fig. 7 shows how GCBR provides a series of cases to support multi-stage decision making. Few previous studies have paid much attention to multi-stage decision making. Smyth, Keane, and Cunningham (2001) described the technique of hierarchical case-based reasoning, which borrows ideas from hierarchical planning and uses a divide-and-conquer strategy to enable the solution of complex problems by reusing multiple cases at various levels of abstraction along an abstract-to-concrete continuum. They employed this technique to design device-control or process-control software for industrial applications. Their focus differs, however, from multi-stage decision making in the real world. To implement a real multi-stage case-based recommender, the target requirements should be rewritten in each stage according to previous actions or responses. The *target-rewrite-mechanism* is a core algorithm applicable to multiple stages of reasoning, as Fig. 8 illustrates.

Fig. 8 illustrates an overall case feature that includes a consumption feature, an accumulation feature, a replacement feature, and a feature for other factors. With the first three, the actions or responses of each stage change the feature values of the next. Performing the recommender for the next stage's recommendations therefore requires rewriting the target features, so the *target-rewrite-mechanism* is able to call the *fet-changecheck-function* to check for any changes necessary. The algorithm then alters these features and generates a new target the next stage of reasoning.

Fig. 9 presents an example involving sightseeing in which the available budget, which is the consumption feature, for travel planning decreases with each stage, while the ever-been-sightseeing factor, which is the accumulation feature, increases accordingly. The algorithm therefore needs to alter the target before performing the next case-recommendation stage. GCBR must first deduct the previous stage's sightseeing entrance fees from the available

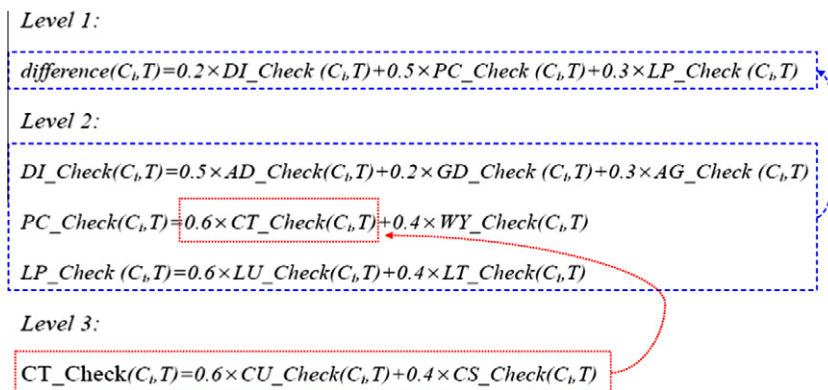


Fig. 4. Example of a similarity evaluation in an HCA problem.

```

For i=1 to n
  For j=1 to m
    if ( level( fet_j^i ) > 1)
      fet_j_check_function = fet_check_rewrite(fet_j);
    end if;
    gap(fet_j^C_i, fet_j^T) = fet_j_check_function Equation (2)
  next j;
  difference (C_i) = sum_{j=1}^m gap(fet_j^C_i, fet_j^T) Equation (3)
next i;
Recommendation_Cases=Reuse(difference,threshold);
(Potential_rules, Sol_Suggestion)=KDD(Recommendation_cases);
    
```

Fig. 5. GCBR's single-stage reasoning algorithm.

```

Function Recommendation_Cases=Reuse(difference,threshold);
  Recommendation_case={};
  For i=1 to n
    If (difference(Ci) < threshold)
      Recommendation_case=Recommendation_case+Ci;
    End if
  Next i;
  If Recommendation_case={};
  /* return the most similar one even if the difference cannot satisfy the
  threshold */
    find C_q having minimal difference in difference(Ci) i=1,2,...,n
    Recommendation_case=Cq;
  endif
End reuse
    
```

Fig. 6. The single-stage reasoning algorithm's reuse algorithm.

budget feature to obtain the new available budget. It also needs to incorporate the site visited in the previous stage into the ever-been-sightseeing feature, as most travelers do not want to repeatedly visit the same sites during a short vacation. In order to provide recommendations for the next stage it should therefore revise the target to recognize the previous stages' actions and responses.

3.3. GCBR improving efficiency via GA

The GCBR algorithm's overall complexity exceeds that of traditional CBR in order to fulfill its reasoning's general and forecasting potential, even with the adoption of the revised CBR (2008). The traditional CBR's reasoning process compares every case in the case base in order to obtain feasible cases to refer to the decision

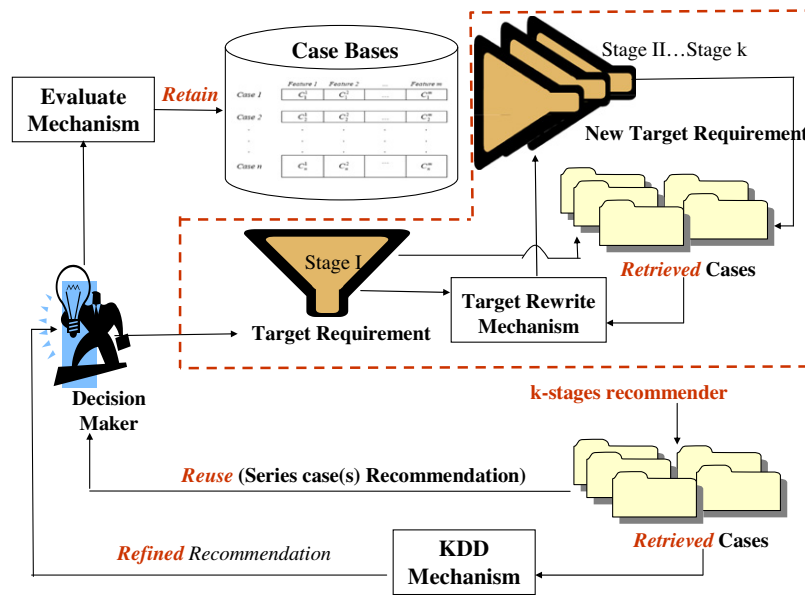


Fig. 7. The multiple stage GCBR process, including KDD.

```

For p=1 to k
  difference( Ci )=Single_Stage_Reasoning (Case-Base, T, threshold),
    i=1,2,...n;

  Recommendation_Cases=Reuse(difference,threshold);
  (Potential_rules, Sol_Suggestion)=KDD(Recommendation_cases);

  For j=1 to m
    If (Type( fetjT )=consumption)
      fetjT = fetjT - Sol_suggestion( fetj )
    elseif (Type( fetjT )=accumulation)
      fetjT = fetj + Sol_suggestion( fetj )
    elseif (Type( fetjT )=replacement)
      fetjT = Sol_suggestion( fetj )
  Next j;
Next k;

```

Fig. 8. The GCBR's multi-stage reasoning algorithm.

```

Function New_Budget= budget_changecheck (view, budget)
If (budget>view.ticket)
  New_budget=budget-view.ticket
else
  New_budget=0
End if
End budget_changecheck;

```

Fig. 9. The sightseeing example's target-rewriting-mechanism in the *budget-changecheck-function* algorithm.

makers. As the problem requiring a decision becomes more complicated, however, the reasoning process is likely to become increasingly time-consuming. As the number of features increases, furthermore, the evaluation of the similarities between the cases and the target takes up an increasing amount of computer memory, particularly if the problem's description is in the HCA style. GCBR's efficiency therefore needs to improve in order to enable it to function well.

Some works have integrated CBR with other artificial intelligence techniques. Juan, Shin, and Perng (2006) combined CBR with

a GA to customize housing plans, Shin and Han (1999) used one to support CBR in order to enhance classification accuracy, and Yang and Wang (2009a), Yang and Wang (2009b) also successfully combined a GA with CBR to accelerate case evaluation. These approaches integrating GAs with CBR have exhibited superior performance. It therefore seems to be a good method of improving CBR efficiency.

$$\text{difference}(C_i) = \sum_t w_t \times \text{gap}(fet_j^{C_i}, fet_j^T), \quad i = 1, 2, \dots, n \quad (4)$$

$$\text{gap}(fet_j^{C_i}, fet_j^T) = \begin{cases} 0, & \text{if no gap} \\ 1, & \text{otherwise} \end{cases} \quad (5)$$

This study also implemented GCBR using a genetic algorithm that expressed the HCA problem using goal programming. GCBR employed the goal gap in Eqs. (4) and (5) as a fitness function. It further regarded the gap between $case_i$ and T as the survival probability and used it in the evolution of the next generation. If $case_i$ has the smallest gap from target, then this study regards it as an outstanding gene, and thereby has a higher probability of survival. It has, furthermore, adopted the robin-wheel selection mechanism to perform GA selection. The higher the survival probability, therefore, the higher the possibility that the gene or genes could persist to the final generation. Finally, the best chromosome, which consists of the fittest gene, represents a series of retrieved cases for the decision makers' reference.

4. An experiment and an illustrated scenario

We conducted an experiment to validate the general GCBR model's efficiency by validating its characteristics. To do this, we designed an HCA problem, implemented the model on GCBR, and compared its experimental efficiency with that of traditional CBR. This section also presents a scenario illustrating a proposed IT certification path to explain the recommender's multiple stages.

4.1. Experiment 1: travel case recommender

We obtained the experimental cases from a free online dataset called *Travel*. Each of the 1024 cases had 14 features and the three

dimensions of metadata structure of traveler, holiday content, and budget. Fig. 10 illustrates this. For each dimension, decision makers could set such hierarchal criteria as budget, travel duration, and hotel accommodation for each dimension for the entire travel program.

Fig. 11 shows how decision makers can input their query target conditions via a web interface. Fig. 11(a) displays the three dimensions. Decision makers can input their query conditions for each dimension and then give a real number, from 0 to 1, to represent the importance of each feature. We standardized these importance weights and used them to evaluate further similarity. Fig. 11(b) shows how clicking on “Budget Dimension” prompts the lower level features of “Travel Duration” and “Hotel Accommodation” to the decision makers. Since it describes the problem in an HCA style, this web interface enables users to reduce target conditions accordingly to make them more specific.

As described above, each feature has its *fet-check-function*. The following function therefore replaces the similarity evaluation.

$$\text{traveler_check} + \text{holidaycontent_check} + \text{budget_check}$$

The *fet-check-rewrite* mechanism then rewrites each dimension.

Level 1:

$$\text{traveler_check} + \text{holidaycontent_check} + \text{budget_check}$$

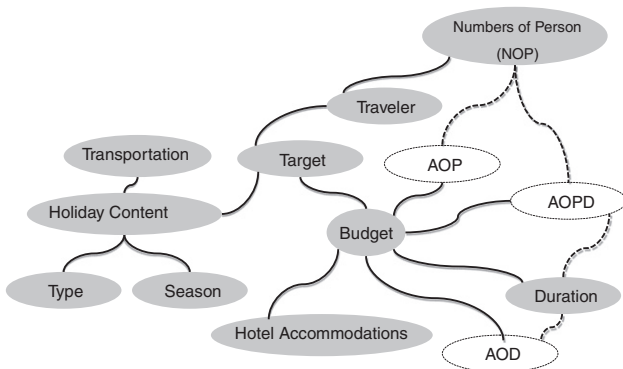
Level 2:

$$\text{traveler_check} = \text{traveler_check} + \text{NOP.check} + \text{AOP.check} + \text{AOPD.check}$$

$$\text{holidaycontent.check} = \text{holiday.check} + \text{type.check} + \text{season.check}$$

$$\text{budget_check} = \text{budget_check} + \text{duration.check}$$

The experiment assumed that the decision maker intended to spend no more than \$2500, with an importance weight of 0.7, that at least two persons would travel on the trip, with an importance weight of 0.4, and that they wanted a vacation focused on active travel, with an importance weight of 0.5. In regard to the budget the experimental decision maker set the trip’s duration at a minimum of seven days, with an importance weight of 0.4, and set the standard of staying at three-star accommodation at the minimum, with an importance weight 0.4. Within the holiday content dimen-



1. NOP means “number of person” attending this travel program.
2. AOP means “average budget of each person”.
3. AOD means “average budget of each single travel day”.
4. AOPD means “average budget of each person in each single day”.

Fig. 10. Meta-data structure of Experiment 1.

sion the decision maker preferred to travel in summer, with an importance weigh of 0.3. We addressed this problem by applying GCBR, using the different crossover rates of $C = 0.3$ and $C = 0.5$ and mutation rates of $M = 0.1$ and $M = 0.01$ for five generations. In an average time of 0.2398 s the program retrieved five cases for the decision maker. Table 2 shows the efficiency comparison.

Fig. 12 illustrates the experiment’s results. The program has standardized the target’s importance weights and displayed them and the convergence time to the decision maker. It has listed all five of the cases it has retrieved on the screen for reference, and the decision maker can click on the button “GCBR Stage II,” which is the data mining process, to obtain more refined information. These target conditions have produced such findings as that the costs range from a minimum of \$988 to a maximum of \$2355, that three persons are apparently the perfect number for such a travel plan, that the optimal trip length is apparently 14 days, and that a car is apparently the best method of travel. Information with this level of detail provides decision makers with a sound basis for revising their actions.

We then tested different parameter combinations. Fig. 13 shows that experiment’s results, which indicate a stable convergence rate at 3–4 generations that was higher with lower crossover rates. The suggested parameters for GCBR are therefore $C = 0.3$ and $M = 0.01$. As Table 2 shows, GCBR can therefore reduce for evaluation approximately 80% more cases than traditional CBR and still provide decision makers with a sufficient number of reference cases.

4.2. Experiment 2: IT certification recommender

IT certification is increasingly important for obtaining employment in the industry, and employers frequently consider it a key screening mechanism. Venator (2006) contended that such certification, as well as educational background, has become a standard for determining applicants’ suitability as IT workers, and Jo (2005) concluded further that IT certification increases women’s career opportunities, particularly in regard to information security. It seems to be a master key for unlocking the doors to job opportunities and career promotion. Almost all IT students have the goal of obtaining it, as do IT workers (Brookshire, 2002). However, the examinations for earning it are interminable, as approximately 200–400 computer-related certifications exist (Zeng, 2004), and the only way to obtain them is to pass the required exams. Their number, furthermore, is continuously increasing, and their content changes by roughly 10% to 15% annually.

Certification exams are therefore a major concern for many IT workers. Even exam veterans have such problems in regard to preparing for and taking them as deciding which ones to take, learning what the current certification is, what the required courses are, and what the restrictions for applying for them are, and finding the optimal method of preparing for them. They need a personalized case-based recommendation mechanism to address these problems (Dolog & Sintek, 2004).

The following scenario illustrates this paper’s proposed multi-stage recommendation mechanism for IT certification. A hypothetical student obtained her information-management master’s degree in 2003 and then served as a database management administrator (DBA) for the ABC software company. She passed the Cisco Certified Network Associate exam, obtained certification, and later obtained Oracle OCA 9i certification. With these qualifications, she submitted a query about receiving a personalized recommendation in order to benefit from others’ examination experiences. Fig. 2 shows how she was able to use the system to establish comparison conditions according to her personal demographic data, capabilities, and learning path. She also considered her work experience and current certification to be possible domain capabilities.

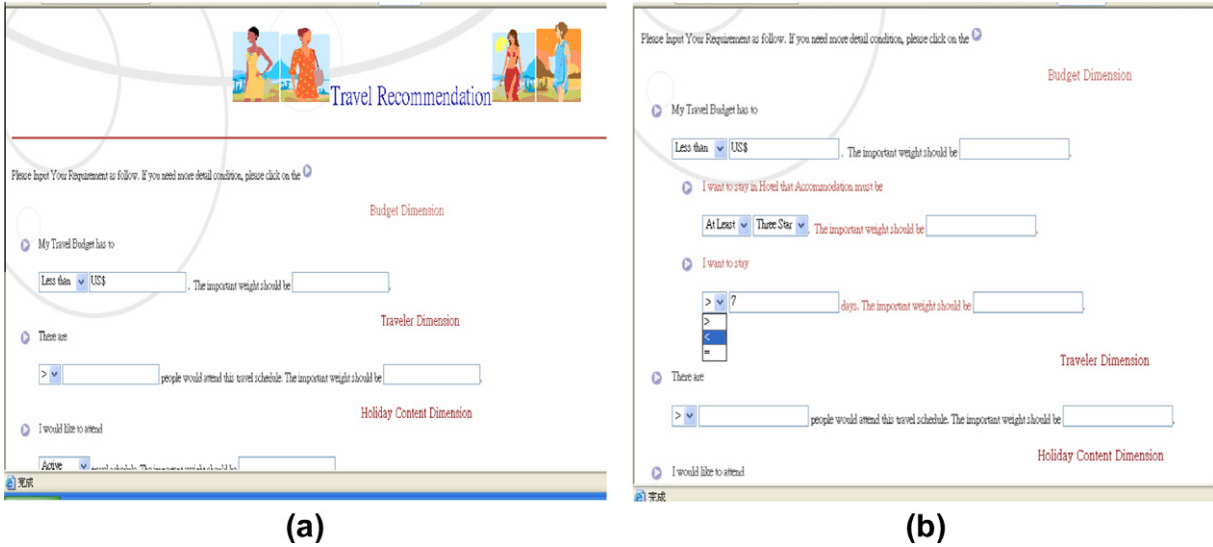


Fig. 11. Demonstration of an HCA case-based recommendation for Experiment 1.

Table 2 Efficiency comparison of the GCBR and traditional CBR in Experiment 1.

Crossover rate	Mutation rate	Cases evaluated		Number of cases suggested
		Amount	Compared to traditional CBR	
C = 0.3	0.01	212	Reduced 79.30%	4
	0.10	190	Reduced 81.45%	3
C = 0.5	0.01	209	Reduced 79.59%	5
	0.10	185	Reduced 81.93%	4

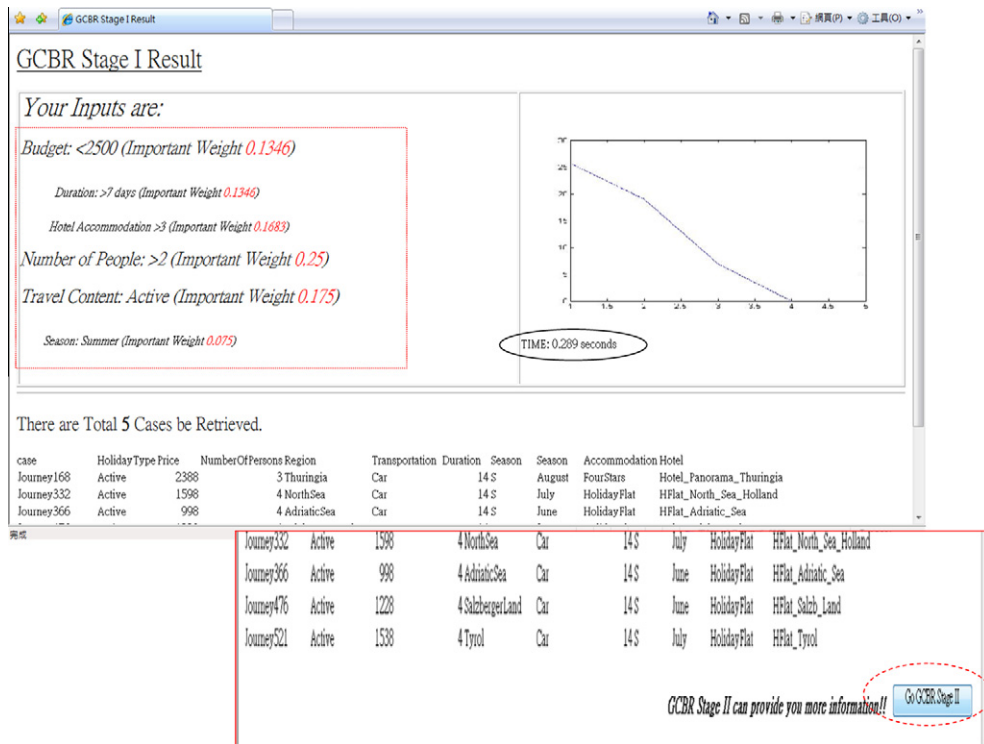


Fig. 12. Demonstration of the Experimental 1 result.

Fig. 14 illustrates the multiple certification paths the system recommended to her in response to these target conditions. The

figure's boldface italic characters represent domain knowledge and IT certification and indicate that the system verified her

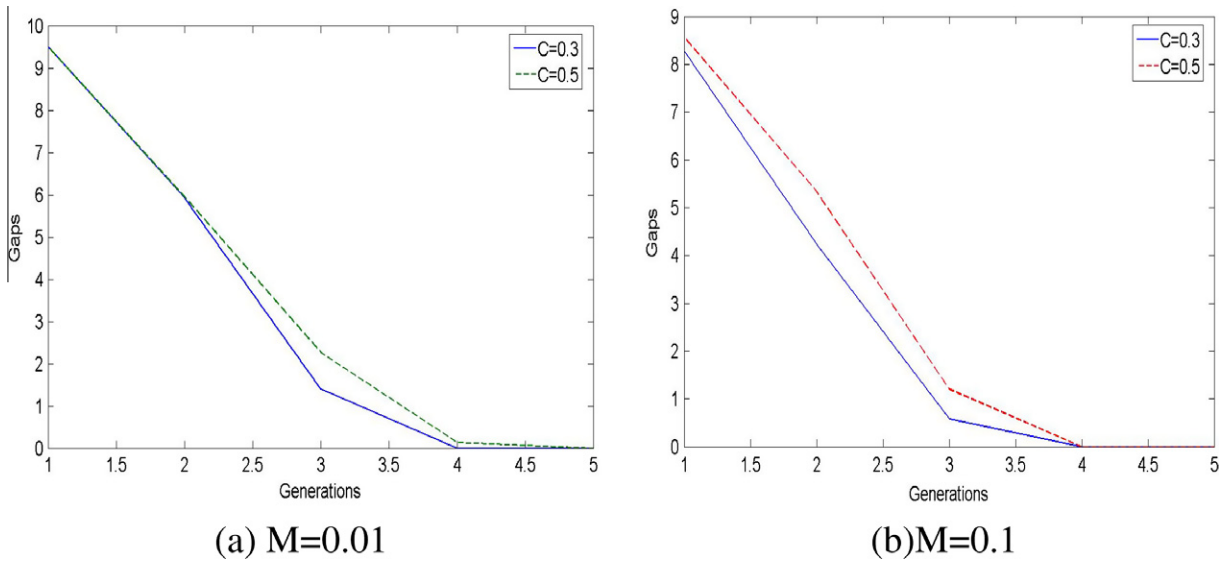


Fig. 13. Convergence of different parameter combinations in Experiment 1.

domain knowledge and the certification she obtained. The connection between domain knowledge and IT certification, furthermore, represents the domain knowledge she needed to pass the certification exam. The recommendation mechanism also offered a sequence of two IT certification exams in case she wanted the system to provide her with IT certification planning advice in two stages.

The system therefore recommended that she obtain the Project Management Professional (PMP) certification first because it required no further preparation, followed by the Oracle OCA 10 g, which is an OCA 9i upgrade that requires more feature-domain knowledge in addition to that of the 10 g, and the Oracle OCP 9i, which is another advanced DBA certification requiring further domain knowledge. The multi-path mechanism arranged the priorities for the certification paths based on their domain knowledge coverage. Fig. 14 shows the eight learning paths to which the system referred her. These are: {OCP 9i → OCA 10 g}, {OCP 9i → PMP}, {OCA 10 g → OCP 10 g}, {OCA 10 g → PMP}, {OCP 9i, OCA 10 g → OCP 10 g}, {OCP9i, OCA10 g → PMP}, {OCA 9i (if failed, test again) → OCA 9i}, {OCA 10 g (if failed, test again) → OCA 10 g}.

She therefore had to consider whether she wished to become a professional DBA, for which an Oracle database certification would be suitable, or to extend her career’s scope, as project management

apparently requires another career plan, so she could decide to work toward PMP certification. The application’s constraints, however, warned her that she was not qualified to take the PMP exam, as the project management institution’s regulations stipulate that all examinees must attend at least 30 h of training classes at an approved institution. She therefore must decide whether to work to fulfill this requirement or apply for some other certification.

From a lifelong learning perspective, recommending a series of related IT certifications to learners would enable them to feel confident that their career planning will match their expectations. Providing them with a choice of a series of certification paths could also increase their confidence about their upcoming certification examinations.

5. Conclusions

This study has proposed an HCA method for describing complicated real-world decision-making problems. Its problem descriptions enable decision makers to clarify problems requiring decisions systematically. HCA is, furthermore, a generalized problem-representation methodology that subsumes the traditional method of describing multiple-objective problems. This paper has also proposed the next generation case-based recommender mechanism GCBR for solving HCA problems. GCBR retrieves feasible cases for reference and then applies the KDD mechanism to provide decision makers with refined information upon which to take action.

This study then found that GCBR can be implemented with a GA algorithm to accelerate the convergence of complex problems that its convergence rate is satisfactory, and that in the experimental case it reduced effort by approximately 80% from that of traditional CBR. It can also provide solutions for problems with multiple stages. This study’s hypothetical scenario illustrated that using it as an IT certification recommender can provide learners with a series of IT a certification path that enables them to feel confident that their career planning will match their expectations. These characteristics enable decision makers to apply the GCBR algorithm as a general case-based recommendation mechanism.

The quality of each case obviously limits what case-based recommender mechanisms can do, so future studies need to address the problem of case cleaning. Other studies need to consider more real-world applications for the algorithm and user satisfaction with GCBR.

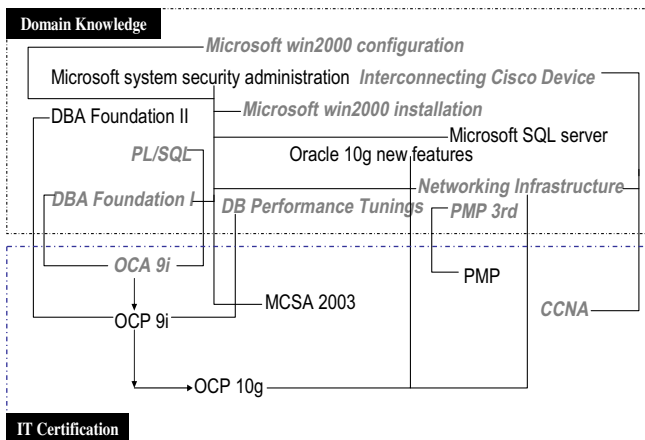


Fig. 14. Multiple recommendation stages.

References

- Aamodt, A., & Plaza, E. (1994). *Case-based reasoning: foundational issues, methodological variations, and system approaches*. AI Communications (vol. 7). IOS Press, No. 1.
- Adomavicius, G., & Kwon, Y. O. (2007). New recommendation techniques for multicriteria rating systems. *IEEE Intelligent Systems*, 22(3), 48–55.
- Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transaction on Knowledge and Data Engineering*, 17(6), 734–749.
- Belecheanu, R., Pawar, K. S., Barson, R. J., Bredehorst, B., & Weber, F. (2003). The application of case based reasoning to decision support in new product development. *Integrated Manufacturing Systems*, 14(1), 36–45.
- Brookshire, R. G. (2002). Information technology certification: Is this your mission. *Information Technology Learning, and Performance Journal*, 18(2), 1–2.
- Cao, L., & Zhang, C. (2007). Domain-driven data mining: A framework. *IEEE Intelligent System*, 22(4), 78–79.
- Chang, C. L. (2005). Using case-based reasoning to diagnostic screening of children with developmental delay. *Expert Systems with Applications*, 28, 237–240.
- Chang, P. C., & Lai, C. Y. (2005). A hybrid system combining self-organizing maps with case-based reasoning in wholesaler's new-release book for forecasting. *Expert Systems with Application*, 29, 183–192.
- Cirovic, G., & Cekic, Z. (2002). Case-based reasoning model applied as a decision support for construction projects. *Kybernete*, 31(5/6), 896–908.
- Coello, C. A. (2000). An updated survey of GA-based multiobjective optimization techniques. *ACM Computing Surveys*, 32(2), 109–143.
- Dolog, P., & Sintek, M. (2004). Personalization in Distributed e-Learning Environments. In *Proceedings of the WWW conference* (pp. 170–179).
- Garrell, J. M., Golobardes, E., Bernado, E., & Llorca, X. (1999). Automatic diagnosis with genetic algorithms and case-based reasoning. *Artificial Intelligence in Engineering*, 13(4), 367–372.
- Golobardes, E., Llorca, X., Salamo, M., & Marti, J. (2002). Computer aided diagnosis with case-based reasoning and genetic algorithms. *Knowledge-Based Systems*, 15, 45–52.
- Jo, S. R. (2005). IT certification: Increasing women's career opportunities. *Certification Magazine*, 12.
- Juan, Y. K., Shin, S. G., & Perng, Y. H. (2006). Decision support for housing customization: A hybrid approach using case-based reasoning and genetic algorithm. *Expert Systems with Applications*, 31(1), 83–93.
- Kolodner, J. L. (1993). *Case-Based Reasoning: Techniques for Enterprise Systems*. San Mateo, CA: Morgan Kaufman.
- Shimazu, H. (2002). ExpertClerk: A conversational case-based reasoning tool for developing salesclerk agents in e-commerce webshops. *Artificial Intelligence Review*, 18(3/4), 223–244.
- Shimazu, H., Shibata, A., & Nihei, K. (2001). ExpertGuide: A conversational case-based reasoning tool for developing mentor in knowledge space. *Applied Intelligence*, 14(1), 33–48.
- Shin, K. S., & Han, I. (1999). Case-based reasoning supported by genetic algorithms for corporate bond rating. *Expert Systems with Application*, 16, 85–95.
- Shiu, C. K., & Pal, S. K. (2001). Case-based reasoning: concepts, features and soft computing. *Applied Intelligence*, 21(3), 233–238.
- Smyth, B., Keane, M., & Cunningham, P. (2001). Hierarchical case-based reasoning integrating case-based and decompositional problem-solving techniques for plant-control software design. *IEEE Transactions on Knowledge and Data Engineering*, 13(5), 793–812.
- Venator, J. (2006). IT certification: Still valuable after all these years. *Education & Careers*, 28–31.
- Waston, I. (1997). *Applying Case-Based Reasoning: Techniques for Enterprise Systems*. San Francisco, CA: Morgan Kaufmann.
- Yang, Q. (2007). Learning actions from data mining models. *IEEE Intelligent System*, 22(4), 79–81.
- Yang, H. L., & Wang, C. S. (2008). Two stages of case-based reasoning – Integrating genetic algorithm with data mining mechanisms. *Expert Systems with Applications*, 35(1–2), 262–272.
- Yang, H. L., & Wang, C. S. (2009a). Recommender system for software project planning – One application of revised CBR algorithm. *Expert Systems with Applications*, 36(5), 8938–8945.
- Yang, H. L., & Wang, C. S. (2009b). Personalized recommendation for IT certification test in e-learning environment. *Journal of Research and Practice in Information Technology*, 41(4), 295–306.
- Zeng, F. (2004). A new approach to integrate computer technology certifications into computer information system programs. In *Proceedings of the 2004 American society for engineering education annual conference & exposition*.