EG0800332

Genetic Algorithms for Case Adaptation

Abdel- Badieeh M. Salem [1] & A. H. Mohamed [2]

[1] *Computer Science Dept, Faculty of Computer and Information Sciences, Ain Shams University, Cairo, Egypt.* absalem@asunet.shams.enu.eg

[2] *Solid State Dept., (NCRRT), Cairo, Egypt.*

ABSTRACT

Case based reasoning (CBR) paradigm has been widely used to provide computer support for recalling and adapting known cases to novel situations. Case adaptation algorithms generally rely on knowledge based and heuristics in order to change the past solutions to solve new problems. However, case adaptation has always been a difficult process to engineers within (CBR) cycle. Its difficulties can be referred to its domain dependency; and computational cost. In an effort to solve this problem, this research explores a general-purpose method that applying a genetic algorithm (GA) to CBR adaptation. Therefore, it can decrease the computational complexity of the search space in the problems having a great dependency on their domain knowledge. The proposed model can be used to perform a variety of design tasks on a broad set of application domains. However, it has been implemented for the tablet formulation as a domain of application. The proposed system has improved the performance of the CBR design systems.

Key words*: *Case Based Reasoning / Genetic Algorithms / Case Adaptation / Optimizing Case Adaptation.

INTRODUCTION

Case based reasoning systems solve problems by reusing the solutions to previously solved problems. Typically, there are differences between a new problem and the previous case so the success of a CBR system often critically depends on its ability to adapt the solution of a previous case to suit a new situation [(1)].

One of the main advantages of CBR is the relative ease of constructing and maintaining systems, especially as a number of commercial CBR tools are available. However, this advantage applies mainly to CBR systems *relying* on retrieval for finding solutions, e.g., many systems for classification and estimation tasks. Although these tools usually also provide some basic facilities for coding adaptation rules, they do not provide means of automating the task of acquiring the adaptation knowledge [(2)].

Adaptation typically requires a significant knowledge engineering effort. This is a common obstacle for both classification and design tasks. However, adaptation for non-classification tasks is a more challenging problem. For instance, a design solution typically consists of multiple properties rather than a single class label, furthermore design solutions are subject to constraints and compatibility requirements which are often difficult to elicit from domain experts. Therefore, there is a clear need for effective knowledge acquisition tools for the design process [(3)].

The proposed system suggests the use of genetic algorithm methodology for case adaptation in an attempt to overcome the difficulty of the adaptation task, as well as to promote intelligence through it.

This paper is organized as follows: Section 2 describes the CBR approach, explains its adaptation task, and the previous learning methods used in automating the adaptation acquisition.

Section 3 represents GA methodology. Section 4 discusses the proposed system. Section 5 deals with the applicability of the suggested system and the results. Section 6 represents the conclusion.

CASE BASED REASONING APPROACH

Case based reasoning is a paradigm based on remembering previous experiences (cases). A new problem is matched against cases in the case base and one or more similar cases are retrieved. A solution suggested by the matching cases is then reused and tested for success. Unless the retrieved case is a close match, the solution will probably have to be revised (or adapted) producing a new case that can be retained. This new solution is stored for future use [4, 5]. Thus, adaptation has been argued that it may be the most important step of case-based reasoning as it is used to modify the past solution(s) to achieve a new solution for the new problem [6]. But, it is often considered as the most difficult task of a CBR system.

- **Acquiring Adaptation Knowledge**

The difficulty of acquiring adaptation knowledge was identified early in CBR research. In most previous work on case-based design, the adaptation of a design case is formalized using knowledge based and heuristic techniques. They require large amounts of domain knowledge have to be incorporated into the system, and all possible adaptation scenarios must be foreseen and recognized, in order for the adaptation to result in feasible solutions [7].

However, overall, little research has been done on automating acquisition or learning adaptation knowledge. Some of this work is introduced by Hanney and Keane to implement a CBR system that estimates property values, and learns adaptation rules from cases stored in its case-base [8]. DIAL also learns adaptation knowledge, in this case for a CBR planning system. It stores previous successful adaptation strategies that combine domain independent abstract adaptation rules with search procedures for domain specific information [9].

Also, introspective learning (IL) has been used to improve the accuracy of the adaptation process. It determines the differences between the rules used to solve the old cases, and combines them to generalize adaptation rules to solve the new problem [10].

In our current work we propose the use of GA to perform design case adaptation. Applying GA in the CBR adaptation is considered one of the knowledge light methodologies. These algorithms have several advantages with respect to most knowledge-based reasoning algorithms: they require less domain knowledge in order to operate, while still producing "feasible" results; they are more dynamic in that they are not limited to describing design cases using a predefined scheme with a fixed set and a fixed number of variables; and they are more flexible in that they can combine pieces from several past experiences in order to solve a new problem, a capability that seems necessary for creative design [11].

Some systems have been used GA for case adaptation. But, they had concerned with some components of system only [2], or used multiple mechanisms inside one system each of them can deal with one excipient only [3]. While, proposed system suggests the use of GA as a single mechanism for adapting all the system components.

GENETIC ALGORITHMS

Genetic Algorithms (GAs) are biologically inspired and have a great deal of potential in scientific and engineering optimization or search problems [(12)]. GAs operate well under conditions where little domain knowledge is known or the search space is increasingly difficult. This fits well in CBR and the adaptation task, as CBR is often used for systems where there is little domain knowledge known [(13)].

For applying a GA to any domain, we must have an initial *population* of individuals (*chromosomes*) which is operated upon, and assigning a *fitness* score which assesses how good a solution the chromosome is to the problem. Highly fit chromosomes are given opportunities to *reproduce* themselves by *cross breeding* with other individuals in the population. New individuals are produced from this cross breeding, known as *offspring*. To produce offspring, GAs apply the genetic operators *crossover* and *mutation*. Crossover is an operator allowing new points in the search space to be tested, where mutation maintains that no point in the search space has zero probability of being explored [(14)]. Offspring share features taken from each parent. Less fit members of the population are less likely to be selected for reproduction, and hence die out of the population due to survival of the fittest. A GA must also have an applied selection scheme (to select chromosomes for reproduction), as well as values for probability of crossover (*Pc*) and probability of mutation (*Pm*) [(15)].

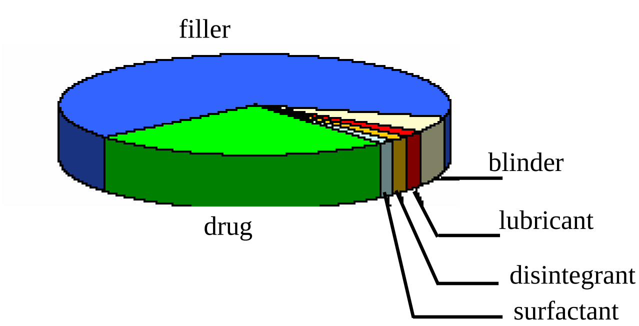
APPLYING A GENETIC ALGORITHM TO CASE ADAPTATION (GA-CBR) FOR THE TABLET FORMULATION

- **Empirical Case Study**

The proposed system developed for optimizing CBR adaptation is generic and can be applied to many design problem domains. In this research we illustrate its use for design task – tablet formulation. However, tablet formulation process has a great importance in real-life, and it can be considered one of the most economic domains especially in recent years.

The design of a new tablet involves identifying inert substances called *excipients* to balance the properties of the drug so that the tablet is manufactured in a robust form, and the desired dose of drug is delivered and absorbed by the patient [(16)] .

Excipients play the role of fillers, binders, lubricants, disintegrants and surfactants in the tablet. Furthermore, a formulation specifies the quantity of each of the added excipients. The difficulty of the formulation task arises from the need to select a set of excipients compatible with the drug, whilst at the same time satisfying a variety of soft and hard constraints as shown in fig. (1) [(17)].



drug: Active ingredient
filler: provide bulk to be large enough to handle or compress.
binder: used as a granulation agent
lubricant: enables tablet to ejected from the die
disintegrant: To rapid breakup and disintegration after administration
surfactant: To aid wetting and dissolution of the drug.

Fig. (1) : The components of the tablet

Tablet formulation practice suggests that the design task is decomposable into subtasks consisting of the choice of excipient component plus determining the excipient amount. It is possible to find a sequence in which the components will be determined, from the most constrained, filler, to the least constrained, surfactant, so that most tablets can be formulated without the need to backtrack in the formulation process.

- **Case Representation and Retrieval of (GA-CBR)**

(a) Case Representation

The first task of CBR is the case representation issue. In our system, design cases are represented using the attribute-value formalism, widely used in supporting design problem solving. New design problems are described using the same formalism: attribute-value pairs that describe requirements of the new design. The representation of the design cases can include more than the design specifications, such as contextual or support information [18].

The feature vector for a case of tablet formulation task contains 26 features: 5 are physical properties of the drug, 20 are chemical properties of the drug with the excipients, and one for the tablet properties.

The cases of formulations can be represented by:

- The problem part:
  - a set of physical features describing the drug (e.g. solubility, dose, mechanical properties, etc)
  - the set of stabilities of the drug with each of the available excipients

These features are: Drug Name, Dosage, Drug Concentration, Drug Solubility, Drug Stability, Drug Yield Pressure.

- The solution part:
  - the chosen filler and its quantity
  - the chosen binder and its quantity
  - and similarly for lubricant, disintegrant and surfactant.

And the tablet properties are: Tablet Diameter, Tablet Weight, Height, Strain Rate Sensitivity.

The features in the problem part are used to identify the cases in the database that are most similar to the new problem's features.

(b) Retrieving Stored Cases of (GA-CBR)

In the retrieval task, the CBR system must determine the relevance of a design case to the current problem-solving situation. That requires matching the attribute-value pairs in the problem specification with those contained in the design cases [19].

Proposed GA-CBR system uses the k-NN nearest neighbor algorithm that has proven its suitability for solving the design problems [18]. It retrieves the k similar cases of the new case. The formulations in the retrieved cases are good first attempts to formulate the new tablet, but it is unlikely that they can be used directly.

(C) Proposed Genetic Algorithm for Adaptation Process

The adaptation stage refines the retrieved formulations by identifying and taking account of differences between the new and retrieved problems. Thus, for example, if the new drug is less soluble we may wish to try more soluble filler, or for a larger dose we may need to use less filler. In this way a formulation from the new tablet is proposed.

Applying the genetic algorithms to optimize the adaptation process of CBR, the proposed GA-CBR algorithm has the following steps:

1- ***Phenotype-to-Genotype Transformation***: A CBR case must be mapped to a GA chromosome. Encoding the retrieved cases (phenotype form) are considered as chromosomes (genotype form) of the initial population of GA. These chromosomes are considered as a seed of the initial population of GA. Each chromosome of the tablet formulation GA is a real-value string of 26 genes.

2- Combination of Individuals: Combination of two individuals (parents) is achieved through genetic crossover operators to produces two offsprings. GA uses PMX crossover. The type of adaptation performed by parents' combination can be considered as a structural adaptation (19). However, the content of the information held in the two offsprings is the same as that in the parents, though in different combinations. But the structure of the offsprings is different from the structure of the parents. Crossover rate is used with a value of 0.8.

3- Modification: Genotype modification is achieved through genetic mutation operators. The type of adaptation performed by case modification is also known as parametric adaptation [19]. The appearance (i.e., the structure) of the offspring is the same as that of the parent. But the content of the offspring is different from the content of the parent: the two phenotypes describe different objects. Mutation rate is randomly applied with a value of 0.5.

4- ***Genotype-to-Phenotype Transformation****:* At a given evolutionary cycle t+1, the population of new genotypes generated through crossover and mutation are retransformed into a population of new phenotypes. The mapping function depends on the representations of genotypes and phenotypes chosen for a given application domain and the relation between them.

5- ***Evaluation of Phenotypes:*** Evaluation has to determine whether a proposed solution looks and/or behaves adequately. This means that it has to analyze the phenotype of a proposed solution. The fitness function, which assigns a fitness value to each proposed solution, is used in this manner.

Fitness function F = R+C

where, F : represents the fitness of a proposed solution
R : represents the number of features of the solution that do not match with features indicated in the problem requirements, and
C : represents the number of domain constraints violated by the solution.

For the tablet formulation problem, there are 18 constraints those must be tested to evaluate the individuals. For example, two main constraints must be considered in the tablet formulation problem: (1) if the drug is insoluble then the filler should be soluble. (2) excipients having high stability with the drug are preferred.

In our layout tablet formulation, each domain-imposed constraint to be verified is represented procedurally, as a function that takes a proposed design as input, analyses it according to the conditions corresponding to the constraint, and returns a yes/no answer indicating whether the proposed design violates the constraint or not.

6- ***Selection***: The proposed GA uses the roulette-wheel selection methodology. As the chromosome that has fittest value has a greater chance to be used in the next generation.

7- ***Termination***: The number of generation used in the GA = 150.

EXPERIMENTAL RESULTS

Our system currently has 161 cases in its case library. In order to perform this efficiency experiment, GA-CBR was run 150 times using 12 cases retrieved from a case base. Then, we compare its results with those obtained from 3 systems: (1) learned k-nearest neighbor [10], (2) adaptation gain system that uses GA for construct the filler, and binder only [2]. (3) C 4.5 that uses the induction learning to learn adaptation rules [20].

Because the design process of tablet formation is a decomposition subtasks process, we can report our findings for each of the excipients separately. Table (1) and figure (2) illustrate a comparison between the average accuracy obtained from the proposed GA-CBR system and the other 3 systems for the excipients.

It is found that, applying GA for CBR adaptation cases reports significant improvements of the average accuracies of all the system's components. For example, the accuracy of determining the filler, and its amount can be concluded as: 66% for C4.5, 69% for learned k-nearest neighbor system, 73% for adaptation gain system, and 81% for proposed GA-CBR.

Table (1): A comparison between the average accuracy obtained from the proposed GA-CBR system and the other 3 systems.

	Filler (%)	**Binders (%)**	**Lubricants (%)**	**Disintegrants (%)**	**Surfactants (%)**
GA-CBR System	81	96	84	82	94
learned k-nearest neighbor	69	75	72	63	81
Adaptation Gain System	73	80	-	-	-
C4.5 System	66	51	68	63	58

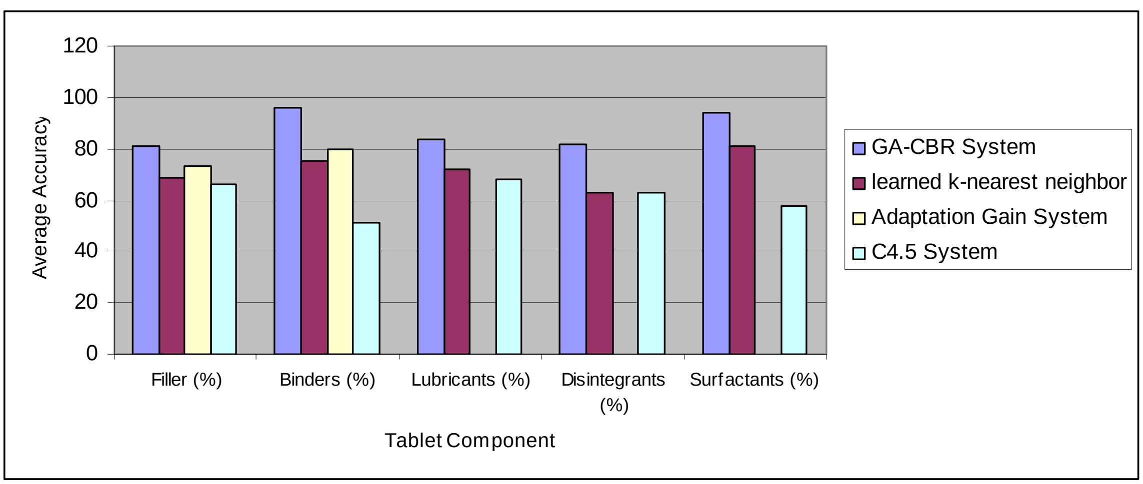


Fig. (2): A comparison of the average accuracy obtained from proposed GA-CBR, and 3 other CBR systems.

It is clear that, adaptation with all the tested algorithms demonstrates substantial improvement over C4.5 system. The performance of adaptation gain is significantly better than learned k-nearest neighbor system, but proposed GA-CBR is the most significantly one that can improve the performance of CBR design process.

However, proposed GA-CBR system has proved its performance when applied for the tablet formulation process as a case of study of design CBR systems.

CONCLUSIONS

Adaptation is the powerful task of CBR systems. It represents the core that enabled CBR to solve similar problems. Knowledge acquisition of adaptation process is a hard, time-consuming task in general. Therefore, there is a need to provide tools to simplify this acquisition process. Many methods have been identified to robust its knowledge acquisition. But, they almostly suffer from some complexity especially for the problems needed a large search space of knowledge, and their needing for a considerable amount of knowledge to have a suitable solutions.

Our research has identified a generic and particularly suited method used to acquire the required case adaptation knowledge for component-based design process. It uses genetic algorithm to optimize the case adaptation task. This process requires a mapping from a case representation to a genotype/phenotype representation, the definition of crossover and mutation operators, and the identification of a fitness function.

The proposed GA-CBR system can improve the performance of the CBR system in many features as:

1- Introduces the uses of genetic algorithm optimization tool to incorporate the domain knowledge into the CBR cycle.
2- Improves the performance of the accuracy of tablet formulation design rather than the other adaptation algorithms.
3- Simplify the required analysis for the adaptation task by using the genetic algorithm that needs low computational complexity in uses.
4- Can be used for designing the low knowledge domain applications.
5- Can be used to determine all the required excipients rather than concern with some of them only achieved by the previous systems (for more details, see [(2)]).
6- Can be used to determine all the required excipients using a single GA adaptation mechanism that can simplify and mange the analysis process rather than using multiple mechanisms inside one system each of them can deal with one excipient only (for more details, see [(3)]).

The proposed system can be applied for many designs in different application domains. It has been applied for tablet formation as a case of study. From the experimental results, GA-CBR system can improve the accuracy of the tablet formulation process rather than the present similar ones.

REFERENCES

(1) F. Oppacher and D. Deugo; In Cercone, N., Gardin, F., eds.: Computational Intelligence III, Elsevier Science Publishers B.V, pp. 103, (1991).

(2) J. Jarmulak, S. Craw and R. Rowe, Proceedings of the Seventeeth IJCAI Conference, pp. 1011, (1998).

(3) N. Wiratunga, S. Craw, and R. Rowe; Proceedings of the 6th European Conference on Case-Based Reasoning, Springer-Verlag, pp. 421, (2002).

(4) L. Purvis and P. Pu; In: Robotica, pp. 285, (1998).

(5) A. Aamodt; AICOM. pp. 39; (1994).

(6) De Silva Garza, A.G., Maher, M.L., Proceedings of Third International Conference on Case-Based Reasoning, pp. 162, (1999).

(7) J. Kolodner; AI Magazine, 12 (2): pp. 52, (1991).

(8) K. Hanney and M. T. Keane; In Proc. 2nd International Conference on CBR, Springer, pp. 359, (1997).

(9) D. B. Leake, A. Kinley, and D. Wilson; In Proc. 13th AAAI Conference, pp. 241, (1996).

(10) Abdel Badieeh M. Salem, A. H. Mohamed; Egyptian Computer Science Journal, Vol. 27, No. 1, pp. 73, (2005).

(11) W. Wilke, I. Vollrath, K.D. Althoff, and R. Bergmann; In 5th German Workshop on CBR, (1997).

(12) II-Seok Oh, Jin-Seon Lee, and Byung-Ro Moon, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, No. 11, November (2004).

(13) J. Jarmulak, S. Craw and R. Rowe; EWCBR, Volume 1898 of LNAI., Springer-Verlag Berling Heidelberg , pp. 136, (2000).

(14) G. Alicia and M. Julie; ECCBR 2004, LNAI 3155, Springer-Verlag Berlin Heidelberg 2004, pp. 198, (2004).

(15) D. Beasley, D. Bull, and R. Martin; University Computing, pp. 58, (1993).

(16) M. Mitchell ; MIT Press, pp. 124, (1996).

(17) J. Jarmulak, S. Craw and R. Rowe, Proceedings 12th IEEE International Conference on Tools with Artificial Intelligence, pp. 376, (2000).

(18) J. Jarmulak and S. Craw; Proceedings of the IJCAI'99 workshop on Automating the Construction of Case Based Reasoners, pp. 258, (1999).

(19) A. Gómez de Silva Garza and M. L. Maher; J. S. Gero (ed.), Artificial Intelligence in Design'00, Kluwer Academic Publishers, Dordrecht. Printed in the Netherlands, pp.193, (2000).

(20) Quinlan, J.R.; In Proc 13th National Conf on AI, pp. 125, (1996).