ELSEVIER Artificial Intelligence in Medicine 18 (2000) 117–132

Artificial Intelligence in Medicine

www.elsevier.com/locate/artmed

A genetic algorithm approach to multi-disorder diagnosis

Staal Vinterbo [a,b,*], Lucila Ohno-Machado [a,b]

[a] *Division of Health Sciences and Technology, Harvard Medical School/Massachusetts Institute of Technology, Cambridge, MA, USA*

[b] *Decision Systems Group, Brigham and Women's Hospital, 75 Francis Street, Boston, MA 02115, USA*

Received 4 January 1999; received in revised form 12 April 1999; accepted 15 May 1999

Abstract

One of the common limitations of expert systems for medical diagnosis is that they make an implicit assumption that multiple disorders do not co-occur in a single patient. The need for this simplifying assumption stems from the fact that finding minimal sets of disorders that cover all symptoms for a given patient is generally computationally intractable (NP-hard). In this paper, we explain the need for performing multi-disorder diagnosis, review previous approaches, formulate the problem using set theory notation, and propose the use of a search method based on a genetic algorithm. We test the algorithm and compare it to another approach using a simple example. The genetic algorithm performs well independently of the order of symptoms, and has the potential to perform multi-disorder diagnosis using existing or newly developed knowledge bases.

Keywords: Differential diagnosis; Multiple disorders; Genetic algorithms

* Corresponding author. Present address: Knowledge Systems Group, Department of Computer and Information Science, Norwegian University of Science and Technology, O.S. Bragstads Pl. 2E, N-7491 Trondheim, Norway. Tel.: +47-73-59-44-80; fax: +47-73-59-44-66.

E-mail address: staalv@idi.ntnu.no (S. Vinterbo)

0933-3657/00/$ - see front matter

PII: S0933-3657(99)00036-6

1. Introduction

Medical expert systems for diagnosis and treatment advice have evolved considerably since the first successful experiments in the early 1960s. MYCIN [14] was perhaps one of the first systems in which concepts from artificial intelligence were used to solve medical problems, namely the diagnosis and treatment of infection in the blood and central nervous system. It was a rule-based system in which no clear distinction between a knowledge base and an inference engine existed. Systems that followed MYCIN, such as Internist-1/QMR [9] and DXplain [2] had knowledge bases that could exist independently of their inference engines. The knowledge base for QMR, for example, has been subsequently used with a Bayesian network inference engine in the QMR-DT project [15,16]. The knowledge bases in QMR and DXplain are composed of quantified relationships between diseases and symptoms. The inference engine is composed of a set of operations that produce proxies for probabilities of single disorders for a particular patient. QMR can suggest additional disorders to improve the cover of the input symptoms, but only conditional on presence of the already chosen disorders in an explanation. Other expert systems have been built for probabilistic calculation [5,3], and used what is called a 'naive' Bayes approach (that is, the systems did produce actual probabilities of disease for a given patient when the assumption of conditional independence among symptoms was correct). In most cases, probabilities for a single disease were produced.

As medical documentation becomes more structured and problem-oriented, problem lists become a central part of patient care. Although not all listed problems can be considered diagnoses (there has been discussion about the finding-diagnosis continuum [6]), it is not rare to see more than one diagnosis in a patient record. In this context, it is possible that a given set of symptoms may be better explained by more than one disorder. We, therefore, argue for the need for expert systems that are able to perform multi-disorder diagnoses.

Examples of multi-disorder approaches are the set theory based approaches by Reggia et al. [13], a variation of this given by Wu [18], and the neural network approach given by Cho and Reggia [4].

The method presented here can be seen as a set theoretical approach that tries to address shortcomings found in the methods mentioned above. The approach uses an extension of a binary relation knowledge base to evaluate sets of disorders. This extension makes the knowledge base compatible with current 'disorder profile' knowledge bases as found in QMR and DXplain. A Disorder set is evaluated conditional on a given set of symptoms, the number and severity of disorders in the set, and the overlap of the set of symptoms that the disorders cause in conjunction with the symptom set given. This evaluation is used by a genetic algorithm for finding sets of disorders that optimize this evaluation.

2. Set-theoretical background

Let $S = \{s_1, s_2, ..., s_k\}$ be a set of symptoms, and let $D = \{d_1, d_2, ..., d_l\}$ be a set of disorders. Let $K \subseteq D \times S$ be a binary relation associating each disorder in D with

symptoms in S that are caused by that disorder. K can then be considered to be a knowledge base. Associate with each symptom $s \in S$, a set of disorders Causedby$(s) = \{d | (d, s) \in K\}$ that cause the symptom, and associate with each disorder $d \in D$, a set of symptoms Causes$(d) = \{s | (d, s) \in K\}$ that it causes, called the *profile* for that disorder. Let also Causedby$(S') = \cup_{s \in S'}$ Causedby(s), Causes$(D') = \cup_{S \in D'}$ Causes(d) for sets $S' \subseteq S$ and $D' \subseteq D$. The task of diagnosis can be stated as the search for a set of disorders that, 'explain' a set of symptoms exhibited by the patient. We follow Reggia et al. in that we view this problem in the context of set theory. They state that a disorder d explains a symptom s if it causes it, i.e., $s \in$ Causes(d). A differential diagnosis can thus be obtained for a set S' of symptoms, by selecting all disorders that explain a symptom in S', i.e., the set Causedby(S'). This set is the least refined explanation we can produce using the knowledge base, and in general not a particularly helpful one. Reggia et al. [13] proposed to find all minimal subsets of disorders such that, by the conjunction of disorders in such a set, all symptoms in a given set S' are explained. These sets they termed 'minimal candidates'. Formally, the set of *candidates* for a given set S' is

$$\text{Cand}(S') = \{D' | D' \subseteq D \quad \text{and} \quad S' \subseteq \text{Causes}(D')\}$$

The set of candidates with no proper subsets in Cand(S') is the set MinCand(S') the set of minimal candidates.

Given the knowledge base

$$K = \{(\mathit{fever}, \mathit{flu}), (\mathit{fever}, \mathit{tb}), (\mathit{fever}, \mathit{hep}), (\mathit{fever}, \mathit{mal}), (\mathit{cough}, \mathit{flu}), (\mathit{cough}, \mathit{tb}),$$
$$(\mathit{cough}, \mathit{asth}), (\mathit{cough}, \mathit{bron}), (\mathit{jaundice}, \mathit{hep}), (\mathit{dyspnea}, \mathit{asth})\},$$

the minimal candidates for the set $S' = \{\mathit{fever}, \mathit{cough}\}$ are:

$$\text{MinCand}(S') = \{\{\mathit{Flu}\}, \{\mathit{tb}\}, \{\mathit{hep}, \mathit{asth}\}, \{\mathit{hep}, \mathit{bron}\}, \{\mathit{mal}, \mathit{asth}\}, \{\mathit{mal}, \mathit{bron}\}\}.$$

Reggia's algorithm, called 'candidate generation', searches for a compact representation of (an approximation of) the set of minimal candidates. Wu presents a restriction of this search space to obtain the same type of representation. A natural search space for finding the minimal set of disorders that explains a set of symptoms is the space of subsets of the known disorders. This search requires at least two measures to be taken into account:

- the size of a set of disorders, and
- how much this set explains the given symptoms.

A useful observation is that an element of MinCand(S') must necessarily intersect Causedby(s) for all s in S'. This leads us to the related concept of a hitting set.

Let U be a universe of elements and let $\mathscr{C} = \{D_1, D_2, ..., D_m\}$ be a collection of non-empty elements from 2^U, the power set of U. A *hitting set* H of $\mathscr{C}$ is a set that has a non-empty intersection with every element in $\mathscr{C}$, i.e. $H \cap D_i \neq \emptyset$ for all $D_i \in \mathscr{C}$. Let HS$(\mathscr{C})$ denote the collection of hitting sets of $\mathscr{C}$. A *minimal hitting set* is an element of HS$(\mathscr{C})$ with no proper subsets in HS$(\mathscr{C})$. Let MHS$(\mathscr{C})$ denote the

collection of minimal hitting sets of $\mathscr{C}$. The problem of finding a minimal cardinality element of MHS($\mathscr{C}$) is often referred to as 'the minimal hitting set problem'. Similarly, we will name the problem of finding MHS($\mathscr{C}$) as 'the minimal hitting sets problem'.

As

$$\text{MinCand}(S') = \text{MHS}(\mathscr{C})$$

for

$$\mathscr{C} = \{\text{Causedby}(s) | s \in S'\},$$

the candidate generation problem can be stated as an equivalent minimal hitting sets problem. For the example knowledge base above and the set $S' = \{fever, cough\}$,

$$\mathscr{C} = \{\{flu, tb, hep, mal\}, \{flu, tb, asth, bron\}\},$$

where the first element $\{flu, tb, hep, mal\}$ are the disorders that cause *fever*, and the second element $\{flu, tb, asth, bron\}$ are the disorders that cause *cough*. The set $\mathscr{C}$ can be seen as a 'generator' for a subset of HS($\mathscr{C}$) that is a superset of MHS($\mathscr{C}$), and can be viewed as a first step towards a solution to the multidisorder problem. This set is composed of all possible sets constructed by taking one element from each set in $\mathscr{C}$. A given collection of sets can be generated by a set of generators, each generator generating a sub-collection of the collection. The collection $\{\{a, b\}, \{a, c\}, \{b, c\}\}$ is generated by the following set of generators: $\{\{\{a\}, \{b, c\}\}, \{\{b\}, \{c\}\}\}$. The first element of this set, $\{\{a\}, \{b, c\}\}$ generates elements $\{a, b\}$, $\{a, c\}$, and the second element, $\{\{b\}, \{c\}\}$, generates $\{b, c\}$. Note that there might exist several generator sets for the same collection of sets. For example, $\{\{\{b\}, \{a, c\}\}, \{\{a\}, \{c\}\}\}$ and $\{\{\{a\}, \{b\}\}, \{\{a\},\{c\}\}, \{\{b\}, \{c\}\}\}$ both again generate the same collection given above. A set of generators can be used as a compact representation of a collection of sets. The candidate generation algorithm searches for such a representation. Note that the last generator set above is essentially equal to our collection, but does not save space if used to represent the collection. It is also evident that the number of possible generator sets from a universe U is much larger than the number of subsets of that universe.

The minimal hitting sets problem, the minimal hitting set problem, and the candidate generation problem are all NP-hard (for a compendium of NP-hard optimization problems, see [1]). Currently, the only way of establishing an optimal solution is to apply an exhaustive search method. We discuss next, the general infeasibility of a brute force approach to this problem, based on the equivalence of the hitting sets problem and the candidate generation problem. Let n be the cardinality of the union U of the elements in $\mathscr{C}$. Then, the cardinality of MHS($\mathscr{C}$) is bounded by $\binom{n}{\lceil n/2 \rceil}$. This bound can be established by considering

that the elements of MHS($\mathscr{C}$) are elements of some anti-chain (incomparable elements) in the lattice 2^U under set inclusion.

3. To candidate generation, and beyond

As we have established that the number of candidates generally is large, additional measures will be needed to select candidates that are the most relevant to our problem. An additional measure we propose, is based on how many symptoms caused by the disorders in a candidate (in conjunction), are missing in the input set (exhibited by the patient). This is motivated by the problem of how to interpret symptoms *not* exhibited by a patient. Consider missing symptoms in the input set S as evidence for the absence of these in the patient, and that missing symptoms in the profile for a disorder (the Causes set) should count against a diagnosis for that disorder. We would then like to find a diagnosis for which the profile corresponds the best to the input symptom set S' exhibited by the patient.

Consider the knowledge base of disorders with associated symptoms as given in Table 1, and symptoms exhibited by a patient (input set) 2, 4, 6 and 18.

The measures of how much the disorder symptom set covers the input symptom set, called input cover, and how much the input symptom set covers the disorder symptom set, called profile cover, are also given in the table. Both disorders A and B have maximal input cover. If we wish to take profile cover into account, disorder B would be ranked first because of highest profile cover. Considering the combination of disorders C and D as a syndrome CD of symptoms in the union of the symptom sets of the disorders C and D, the syndrome has input cover 4/4 and profile cover 4/7. If we care more about profile cover than finding the minimal number of disorders in the diagnosis, we would rank the syndrome CD above disorder B.

Collecting the three measures, namely cardinality (number of disorders). profile cover and input cover into one, we can construct the following function describing the value of a disorder set D' conditional on a symptom set S' that needs to be explained:

$$v(D', S') = \delta_1 \frac{|D| - |D'|}{|D|} + \delta_2 \frac{|S' \cap \text{Causes}(D)|}{|\text{Causes}(D)|} + \delta_3 \frac{|S' \cap \text{Causes}(D)|}{|S'|}$$

Table 1
Disorders and corresponding symptoms with measures of how much disorder symptom set covers the input symptom set (input cover), indicated by bold numbers in the table, and how much the input symptom set covers the disorder symptom set (profile cover)

Disorder	Symptoms	Input cover	Profile cover
A	**2 4 6** 7 9 13 17 **18** 21	4/4	4/9
B	**2 4 6** 11 17 **18** 21 23	4/4	4/8
C	17 **18**	1/4	1/2
D	**2 4 6** 8 29	3/4	3/5

The first term rewards smaller sets, the second rewards profile cover, and the third term rewards input cover. The values δ_1, δ_2, and δ_3 let the user assign weights to these terms individually.

The knowledge base essentially contains three sets, D, S and K. A natural way of extending this base into an extended base K^+ is by the following three mappings,

$$\alpha : D \times S \to \mathbb{R},$$

$$\beta : D \to \mathbb{R},$$

$$\gamma : S \to \mathbb{R}.$$

There are many ways of defining and interpreting this extension. One such interpretation might be that the mapping α represents the importance of a given symptom for diagnosing a disorder, or causal strength; β represents the severity of a disorder, or the urgency for intervention; and γ represents a measure of the importance of explaining the symptom. Internist-1/QMR has similar notions; α could correspond to the 'evoking strength' or 'frequency weight' measure, or any combination thereof, γ could correspond to the 'import' value of the symptom. The mapping β has no direct correspondence in neither QMR or DXplain. In the case where we cannot find nor review all possible solutions, β can be used to concentrate the search on disorders that should not be missed.

When considering the incorporation of the knowledge base extension into the three measures we use to control the search, we try to maximize the values given by the mappings for each measure. Intuitively, given an interpretation of the mappings as above, we want to

- find a minimal number of maximally severe disorders,
- choose the disorders for which the input symptoms are the most relevant, and
- take the importance of explaining the individual symptoms into account.

If we convert the β benefit score into a penalty score β^*, we can use use the sum of the individual β^* values of elements of a set to represent a cost of that set. Formally,

$$\beta^*(D') = \sum_{d \in D'} \beta^*(d)$$

The term

$$\frac{\beta^*(D) - \beta^*(D')}{\beta^*(D)}$$

then rewards minimal sets of maximally severe disorders. The function β^* can be constructed in a similar way, as,

$$\beta^*(d) = \frac{\beta_{\max} - \beta(d)}{\beta_{\max}}$$

We can control how much we want the most severe disorder to contribute to the value of the set by choosing a suitable ε in $\beta_{\max} = \varepsilon + \max_{d \in D} \beta(d)$. Note that

for $\beta = 0$ and a non-zero ε, the value of $\beta^*(D')$ becomes the cardinality of D', $|D'|$.

If multiple disorders in a set contribute the same symptom to the conjunctive causal symptom set, how should we compute the α value for that symptom and that syndrome? A simple solution is to take the maximal value for the contributing disorders and the symptom. Formally,

$$\alpha(D', s) = \max_{d \in D'} \alpha(d, s)$$

We then further define

$$A(D', S') = \frac{\sum_{s \in S' \cap \text{Causes}(D')} [\alpha(D', s)\gamma(s)]}{\sum_{s \in \text{Causes}(D')} [\alpha(D', s)\gamma(s)]}$$

$$B(D', S') = \frac{\sum_{s \in S' \cap \text{Causes}(D')} [\gamma(s)]}{\sum_{s \in S'} [\gamma(s)]}$$

$A(D', S')$ represents the adjusted profile cover, and $B(D', S')$ the adjusted input cover. Assembling the parts from above, we arrive at our adjusted value function

$$v(D', S') = \delta_1 \frac{\beta^*(D) - \beta^*(D')}{\beta^*(D)} + \delta_2 A(D', S') + \delta_3 B(D', S')$$

4. A genetic algorithm approach

Genetic algorithms ([7]) are function value optimizing algorithms, i.e., searching for input values that maximize the output value of a function called the 'fitness function'. Our problem can be seen as the optimization problem of the disorder set value function v conditional on the set S'.

4.1. A review of a genetic algorithm

One way of perceiving genetic algorithms is as a search through a space of potential solutions. A collection of potential solutions, a population, is iteratively refined so as to optimize the fitness measure of the population. The fitness measure of the population is defined using fitness function values of its individuals. Holland [7] encoded individuals in the populations as bit-vectors, and modeled the refinement after Darwinian evolution using operations such as *crossover* and *mutation* inspired from genetics. Crossover is a 'mating' operation, usually taking two individuals from the population as parents and producing two offspring. In its simplest form, namely one-point crossover, it chooses a random element position for splitting the parents and assigns the offspring one part from each parent. This can be seen in Fig. 1 where the vertical bar indicates the splitting position.

parents	offspring
(01001|001)	(01001|101)
(11100|101)	(11100|001)

Fig. 1. One point crossover.

Mutation in its simplest form is just the 'flipping' of a random bit in the individual and is done to ensure that no solution is unreachable. The individuals are often called chromosomes, and the bits that make up the chromosomes are often called genes. The selection of individuals to undergo these operations was done stochastically in a way to favor the fitter individuals, thus giving it the Darwinian flavor. A trait of many problems is that the value of the presence of a particular gene in a chromosome is dependent on the presence of other certain genes, which suggests that the performance of a genetic algorithm might be dependent on the ordering of the genes in the chromosome. This problem is referred to as the 'linkage problem'. Holland proposes an explicit *inversion* operator that permutes the order of the genes in the chromosome without losing the semantics of the chromosome by keeping track of the original positions of the genes. Another issue in genetic algorithm design is premature convergence to local maxima. This problem can be approached by the use of fitness value scaling.

A possible implementation of a genetic algorithm is presented in Fig. 2, and consists of the following parts (see indicated genetic algorithm literature for detailed discussions of the parts listed below):

- `initializePopulation()`: constructs the initial population.
- `evaluate(P, Keep)`: associates with each individual D' in the population P, the value $F(D')$, the fitness function. It also attempts to add the evaluated elements to a fixed size, sorted set `Keep` of the fittest individuals encountered so far.
- `stop(P, generation)`: the stopping criterion. If no changes appear in. either `Keep` or in the average fitness of the population for a predetermined number of generations, `stop` returns `true`.
- `scaleFitnessValues(P)`: applies Boltzmann-scaling of the fitness values of the individuals in the population to avoid rapid convergence to local maxima of the fitness function.
- `selectParents(P)`: selects three sets of elements that will be used to produce offspring by application of either crossover, mutation or inversion These sets are disjoint, and are sampled using fitness proportional roulette wheel sampling with multiple markers (also known as stochastic universal sampling).
- `do`*X*`(ParentSets[i])`: produces a set of offspring by applying the genetic operation *X* (crossover, mutation or inversion) to the set of parents `ParentSets[i]`.
- `recombine(P, OffspringSets[1..3], ParentSets[1..3])`: produces a new population by recombining the sets of parents, the sets of offspring and the old population. A sample of elements obtained by inverse fitness proportional stochastic universal sampling, is replaced by the offspring. Also the fittest individual of the previous generation is propagated to the new population (this is called elitism).

```
P ← initializePopulation()
Keep ← ∅
evaluate(P, Keep)
Generation ← 1
while(not stop(P, Generation)) do
   P ← scaleFitnessValues(P)
   ParentSets[1..3] ← selectParents(P)
   OffspringSets[1] ← doCrossover(ParentSets[1])
   OffspringSets[2] ← doMutation(ParentSets[2])
   OffspringSets[3] ← doInversion(ParentSets[3])
   P ← recombine(P, OffspringSets[1..3], ParentSets[1..3])
   evaluate(P, Keep)
   Generation ← Generation + 1
done
```

Fig. 2. Pseudocode for the genetic algorithm.

For a more in depth discussion of genetic algorithms and how and why they work, see [7,8,10].

The parameters given to the algorithm are the sizes of the parent sets, the fitness function, the parameters of the Boltzmann scaling function, the number of generations to wait before stopping, and the size and additional criterion for retainment in `Keep`.

4.2. Application to multi-disorder diagnosis

For the problem of multi-disorder diagnosis, a straightforward choice of population is a set P of disorder sets, encoded as bit-vectors, where each bit indicates the presence or absence of an individual disorder in the set. We will use the value function v, as defined in Section 3, as our measure of fitness.

Let A, B be subsets of U. If $A \supseteq B$ then $\mathrm{MHS}(\{A, B\}) = \mathrm{MHS}(\{B\})$. Thus $\mathrm{MHS}(\mathscr{C}) = \mathrm{MHS}(\mathscr{C}')$, where $\mathscr{C}'$ is the collection resulting from the elimination of all supersets from $\mathscr{C}$. In terms of symptoms to explain, this property allows us to remove a symptom s' from the input set for which the input set contains a different symptom s, such that $\mathrm{Causedby}(s) \subseteq \mathrm{Causedby}(s')$. If $\mathrm{Causedby}(s) = \mathrm{Causedby}(s')$, the one with the higher γ value should be kept. We apply this reduction before the genetic algorithm.

We used the algorithm implementation presented in Fig. 2. The population initialization can be done either completely randomly, or by random individuals generated by $\mathscr{C}$ (as explained in Section 2). The latter initialization method guarantees that the algorithm finds a set of disorders that explains all the symptoms, and is the initialization method chosen in this application. An additional criterion for a set to be considered for retainment in the `Keep` list is that the set has

an input cover above a certain predefined threshold. Relaxing the input cover requirement allows candidates to omit explanations of the symptoms with lower values of γ, i.e., symptoms that are deemed not so important to explain.

Thus, a diagnostic device, Diag (K^+), takes as input a set of symptoms and uses the method presented above to generate a list of differential diagnoses and their fitness function values. These associated values are then used to rank the differential diagnoses, fittest first.

5. Experiment

Cho and Reggia present in [4] a neural network strategy for dealing with multiple disorder diagnosis. They apply a special learning rule for back-propagation networks to enable what they call 'multiple-winner-takes-all' behavior of the network, i.e., the network can perform multiple disorder diagnosis. The knowledge base of brain damage locations and corresponding symptoms they used can be seen in Tables 2 and 3. Our experiment is a superset of the one presented in [4]

Although a knowledge base extended by the mappings presented in Section 3 was used in our experiment, all mappings were made to map to the value 1, thus in effect giving no extra information beyond what is contained in Tables 2 and 3. We selected 211 sets of disorders to create test cases from. These disorder sets were

- all sets of size one (16 sets),
- all sets of size two (120 sets), and
- random sets for each of sizes three, four and five (75 sets).

Table 2
Disorders and corresponding symptoms they cause (symptom names can be found in Table 3)

No.	Disorder	Symptoms
1	Left medial medulla	2 5 21
2	Right medial medulla	1 6 20
3	Left lateral medulla	15 17 18 20 23 24
4	Right lateral medulla	16 17 19 21 22 25
5	Left medial pons	2 4 6 7 9 13 17 18 21
6	Right medial pons	1 3 5 8 10 14 17 19 20
7	Left lateral pons	3 7 15 17 18 21 23 24
8	Right lateral pons	4 8 16 17 19 20 22 25
9	Left midbrain	2 4 6 11 17 18 21 23
10	Right midbrain	1 3 5 12 17 19 20 22
11	Left cerebellum	17 18
12	Right cerebellum	17 19
13	Left frontal lobe	2 4 6 8 29
14	Right frontal lobe	1 3 5 7
15	Left parieto-temporal lobe	21 23 25 27 28
16	Right parieto-temporal lobe	20 22 24 26

Table 3
Symptom numbers as found in Table 2 with their corresponding names

Nos.	Symptoms
1, 2	Left, right hemiparesis
3, 4	Left, right facial paresis
5, 6	Left, right tongue paresis
7, 8	Left, right gaze palsy (conjugate)
9, 10	Left, right internuclear opthalmoplegia
11, 12	Left, right 3rd nerve palsy
13, 14	Left, right 6th nerve palsy
15, 16	Left, right Horner's Syndrome
17	Nystagmus
18, 19	Left, right hemiataxia
20, 21	Left, right touch proprioception impairment
22, 23	Left, right pain-temperature impairment
24, 25	Left, right facial sensory impairment
26, 27	Left, right hemianopsia
28, 29	Sensory, motor aphasia

For every set of disorders D' created this way, the set of symptoms Causes(D'), the union of the symptoms the individual disorders in D' cause, was given to the diagnostic device as input. The highest ranking differential diagnosis given by the device was then compared to D'. For all 211 applications, the highest ranking disorder set was a subset of the set D' that was used to create the input (a proper subset in 44 cases), and did explain all the input disorders. In eight cases equivalent disorder sets to the highest ranking set, were also found (these have the same cardinality and produce the same symptoms). Cho's and Reggia's neural network, in contrast, did not produce perfect answers for disorder sets of size two in a number of cases.

The average number of generations evaluated by the genetic algorithm was 89.7 ± 74.1, out of which the ten last are stopping-criterion generations. The average number of fitness function evaluations was 1445.6 ± 1211.5 for a fixed population size of 50 and biases $\delta_1 = 0.05$, $\delta_2 = 0.5$ and $\delta_3 = 0.45$. The mutation, crossover and inversion rates were set to 0.05, 0.3 and 0.1, respectively. The `keep` list threshold was set to 1, thus only diagnoses explaining all symptoms were accepted. The entire experiment (all 211 searches) took 120 s real time on a 233 MHz Pentium II running NT 4.0 with 64 MB RAM.

6. Discussion

We have presented a genetic algorithm that incorporates an extended knowledge base to search for the highest ranking differential diagnoses. The ranking is based on severity of disorders, the strength of relationships between disorders and symptom, and the importance of the symptoms.

A diagnostic system was built and tested on a specific problem. It exhibited perfect performance. This experiment was a superset of an experiment undertaken by Cho and Reggia for a neural network approach to multi-disorder diagnosis presented in [4].

6.1. Uni- versus multi-disorder diagnostic approaches

The methods mentioned in this paper can in general be divided into two groups: 'uni-disorder', and 'multi-disorder'. The difference lies in the focus of the methods, whether they are geared towards explaining all the symptoms with one disorder, or a set of disorders. QMR and DXplain fall into the uni-disorder category, although QMR has a limited multi-disorder feature. The combination of uni-disorder models, much like uni-disorder systems, will fail when success is dependent on synergy achieved by combinations of individual choices, each of which would be considered poor in isolation. An example is given in Table 4.

There are four disorders, presented with profiles and values given by the model. In this example, the values represent the input cover of each disorder. It is impossible to say, based only on these values, which combination, if any, will maximize the input cover. One could argue that a combination of disorders that have large values will improve our explanation. As we can see in this simple example, however, the lowest value disorder, number four, is the only one that *has* to be in a differential diagnosis that has maximal input cover, as it is the only one that covers symptom s_4.

The methods presented in [13,18] can both be considered being 'multidisorder' approaches, and both search for generators for the set of MinCand(S'), or an approximation thereof. Both methods are dependent on the ordering of input symptoms. Additionally, as indicated in Section 2, the number of candidates is potentially very large. This means that, even if we had a set of generators that is relatively small, we would still have to generate an evaluation value for all sets to be able to rank them. As shown in Section 2, the space of generators for collections of subsets of a universe is larger than the space of subsets of this universe. Wu decomposes the search space by the use of symptom clustering and argues empirically that his method is more efficient than the candidate generation method, by

Table 4
Example for the failure of combinations of uni-disorder systems[a]

s_1	s_2	s_3	s_4	Disorder	Value
1	1	0	0	1	0.5
0	1	1	0	2	0.5
1	0	1	0	3	0.5
0	0	0	1	4	0.25

[a] The table denotes presence or absence of symptoms in numbered disorders, together with a fictitious value (incidentally the input cover) assigned to each individual disorder profile for a match with the input symptom pattern s_1, s_2, s_3, s_4 all present.

comparing running times for a set of diagnostic problems. This argument would have been strengthened by controlling other factors such as solution quality, and number of symptoms and disorders in consideration.

The method presented here is a more direct multi-disorder approach, as the search space is the power-set of D, and in that it is symptom presentation order independent. The random element of the approach also, at least theoretically, ensures us that we do not exclude any potential solution from possible consideration.

6.2. Tailoring the search

The parameters δ_1, δ_2, and δ_3 allow the user to tailor the search according to needs and decisions about how to interpret the knowledge base.

The first term in the evaluation function $v(D', S')$, controlled by δ_1, drives the search towards fewer, more severe disorders.

When considering the match of two sets of symptoms, S' and Causedby(D'), the elements in Causedby(D') $- S'$, i.e., the symptoms caused by a disorder set, but not found in the patient, can be taken as evidence against the disorder set, or not. A higher δ_2 value will count the absence of the symptoms as stronger evidence against the disorder set, a value of zero will disregard these symptoms in the valuation. A generalization relation (partial order) on disorder sets using our knowledge base can be defined as follows: A disorder set D_1 is more general than another set D_2 if the set of symptoms caused by D_2 is a subset of the set of symptoms caused by D_1, i.e. Causes(D_2) $\subseteq$ Causes(D_1). The δ_2 value can then be interpreted as a control on how specific we want the disorder sets to be. If a concern is the masking of more specific disorders in a resultant differential diagnosis, a secondary search in the disorder generalization relation for more specific disorders with adequate input cover can be undertaken for each disorder in the differential diagnosis, after the primary search described above.

A lower δ_3 value will allow the search to include solutions that explain a certain fraction of the symptoms. By retaining a `Keep` list for each of several symptom explanation thresholds, the user can collect solutions for each of these thresholds in one run.

Similar to the generalization relation of disorders, a symptom set S_1 can be viewed as more general than another symptom set S_2 if the set of disorders that cause S_2 is a subset of the set of disorders that cause S_1, i.e., Causedby(S_2) $\subseteq$ Causedby(S_1). The subset relation can be constructed in sub-quadratic time (see among others [12].) If the input symptom set contains a symptom that is more general than another, the superset elimination from $\mathscr{C}$ described in Section 4.2 results in a deletion of the more general symptom from the input set. The justification of deleting generalizations of symptoms is that we are looking for more specific disorders in the explanation.

There might be binary relations on disorders or symptoms that we want to include in the search. QMR uses a binary relation on the set D of disorders in its knowledge base to search for related disorders to present together with the

disorders already found. If we wish to award disorder sets that contain related disorders, we can use a mapping $\rho: D \times D \to \mathbb{R}$, that reflects the relational strength, in our evaluation function by adding a term, e.g., $\Sigma_{d_1 \in D'} \Sigma_{d_2 \in D'} \rho(d_1, d_2)$. Other binary relations on disorders or symptoms can be included in the search in a similar way.

Information about a pathognomonic or a disorder-disqualifying symptom for disorders can be included in the search by 'locking' the corresponding disorder bit in the individuals in the genetic algorithm population to either one or zero, respectively.

The knowledge base featured in this paper, like the knowledge bases of the other approaches, can be thought of as a weighted graph reflecting relations in the knowledge base. If this graph has more than one component, the standard principle of divide and conquer should be considered, i.e., partitioning the input symptoms corresponding to graph components, and treating them as separate problems. The decomposition can be done by using standard graph-theoretical methods found in most books on computer algorithms.

6.3. Other applications

The method presented in this paper is essentially domain independent. An example of another application domain could be scheduling and/or resource allocation. Consider ordering scientific journals for a research group. The set S could be members of the group, D the different journals available, α priorities of journals each member wants ordered, β^* the cost of ordering a particular journal, and γ could reflect the priority order in the group. The goal would then be to find a collection of journals that minimize cost, and grant at least one wish of every member of the group, preferably those with higher priority.

Another application example could be a search for documents in a keyword-document index. The goal would be to find minimal cost sets of documents that would cover all, or many keywords given as input. The α, β, and γ mappings could reflect frequency of occurrence of a keyword in a document, importance of the document, e.g., citation points for scientific papers, and importance of finding the keyword.

6.4. Limitations and future work

Szolovits and Pauker [17] criticize the bipartite disorders and symptoms model because of the inability to deal with, among others, interactions among disorders. They argue that, since the recording of all possible interactions is infeasible, a generative theory of interactions must be included to solve this deficiency. Patil argues strongly [11] for the inclusion of causal pathophysiological knowledge in a diagnostic system, but he acknowledges the need for ability to use associative relations between disorders and findings as well. The approach presented in this paper can be used in a stand-alone manner for domains where the simple disorder/symptom association knowledge base is either sufficient, the only knowledge base

available, or the only feasible knowledge base. The approach could also be used as an 'agent' in a larger, more complex system.

The knowledge base presented and used here is not large enough to evaluate the full potential of the method proposed: the ability to produce acceptable solutions in large, real-world problems involving a large knowledge base, and many input symptoms. Although it may not be reasonable for a human user to input hundreds of findings for a single case, with the increasing use of automated data collection, it is possible that a computer-based system could provide such information. The testing of the method in such a setting is still necessary and will be investigated.

Acknowledgements

We thank Peter Szolovits for helpful discussions, and Sungzoon Cho for permission to use tables from [4]. This work was funded by grant R29 LM06538-01 from the National Library of Medicine and in part by project grant 107409/320 from the Norwegian Research Council.

References

[1] Ausiello G, Crescenzi P, Gambosi G, Kann V, Marchetti Spaccamela A, Protasi M. Approximate solution of NP-hard optimization problems. Springer-Verlag, 1999.
[2] Barnett GO, Chimino JJ, Hupp JA, Hoffer EP. DXplain. an evolving diagnostic decision-support system. J Am Med Assoc 1987;258(1):67–74.
[3] Bergeron B. Iliad: a diagnostic consultant and patient simulator. MD Comput 1991;8(1):46–53.
[4] Cho S, Reggia JA. Multiple disorder diagnosis with adaptive competitive neural networks. Artif Intell Med 1993;5:469–87.
[5] de Dombal FT. Computer-aided diagnosis and decision-making in the acute abdomen. J R Coll Physicians London 1975;9(3):211–8.
[6] Greenes RA, McClure RC, Pattison-Gordon E, Sato L. The findings–diagnosis continuum: implications for image descriptions and clinical databases. In: Proceedings of the sixteenth Annual Symposium on Computer Applications in Medical Care, 1992:383–7.
[7] Holland JH. Adaption in Natural and Artificial Systems. Ann Arbor: University of Michigan Press, 1975.
[8] Michalewicz Z. Genetic Algorithms + data structures = evolution programs. New York: Springer Verlag, 1992.
[9] Miller RA, McNeil MA, Challinor SM, Masarie FE, Myers JD. The Internist-1/Quick Medical Reference project-status report. Med Inform 1986;145:816–22.
[10] Mitchell M. An Introduction to Genetic Algorithms. Cambridge: MIT press, 1996.
[11] Patil RS. Causal reasoning in computer programs for medical diagnosis. Comput Methods Programs Biomed 1987;25:117–24.
[12] Pritchard P. A simple sub-quadratic algorithm for computing the subset partial order. Inform Process Lett 1995;56:337–41.
[13] Reggia JA, Nau DS, Wang PY. Diagnostic expert systems based on a set covering model. Intl J Man-Machine Stud 1983;19:437–60.
[14] Shortliffe EH, Davis R, Axline SG, Buchanan BG, Green CC, Cohen SN. Computer-based consultations in clinical therapeutics: explanation and rule acquisition capabilities of the MYCIN system. Comput Biomed Res 1975;8(4):303–20.

[15] Shwe MA, Middleton B, Heckerman DE, Henrion M, Horvitz EJ, Lehmann HP, Cooper GF. Probabilistic diagnosis using a reformulation of the INTERNIST-1/QMR knowledge base. I. The probabilistic model and inference engine. Methods Inform Med 1991;30:241–55.
[16] Shwe MA, Middleton B, Heckerman DE, Henrion M, Horvitz EJ, Lehmann HP, Cooper GF. Probabilistic diagnosis using a reformulation of the INTERNIST-1/QMR knowledge base. II. Evaluation of diagnostic performance. Methods Inform Med 1991;30:256–67.
[17] Szolovits P, Pauker SG. Categorical and probabilistic reasoning in medicine revisited. Artif Intell 1993;59:167–80.
[18] Wu TD. A problem decomposition method for efficient diagnosis and interpretation of multiple disorders. Comput Methods Programs Biomed 1991;35:239–50.