A Solution Space approach to handling constraints in Case-Based Reasoning: The Unified Problem

Fei Ling Woon, Brian Knight, Miltos Petridis

University of Greenwich, School of Computing and Mathematics, Park Row, Greenwich, London SE10 9LS UK

{ b.knight, m.petridis} @greenwich.ac.uk

Abstract. This paper aims to answer the following research question: "**Can the standard CBR models be generalised to a unified (problem : solution) space to allow flexible query modes?"** In the standard Case-Based Reasoning (CBR) model, a case is represented as a <problem, solution> pair. The problem space and solution space are treated as separate, and nearest neighbours are retrieved using a metric defined on the problem space. The standard method applies to domains where the similarity assumption is valid: that cases which are near in the problem space are also near in the solution space. This presumes that a metric is also defined in the solution space. In this paper a generalisation of the standard CBR retrieval method, which integrates solution space and retrieval space into a single query space is proposed. The retrieval method is proposed by means of the concept of nearest neighbours to a constraint region. It is shown that the standard CBR retrieval is a special case of this more general model. The advantages of the general model are explained in connection with its more general query modes. Whereas the standard model is only queryable by specification of problem "inputs", the more general model is capable of retrieving general queries on the unified problem-solution space. The advantages of this flexible query form are explained in the paper, by means of a variety of illustrative examples.

1. INTRODUCTION

The motivation for this paper is to give a basis for enhancing the usability of CBR tools. The standard CBR tool requires a user to specify given inputs (from the "problem" space), and retrieve near cases which present outputs (from the "solution space). However, in many cases the user would like to query the case base in a more flexible way. Woon *et al.* (2003b, 2003c) cite examples where the user might want to specify some output values as well as target inputs, as when an engineer wants to find input parameters for a desired output. Also, it may be desirable to add constraints to the CBR system, according to circumstances. For example, a design engineer might want to specify that some design components are unavailable at the time. In the example of Section 5.2 below, a constraint is added to the CBR query, to represent an obstacle in the path of a projectile.

The idea of this paper is to treat cases just as data points in the unified space (problem + solution space). In this manner, the CBR retrieval mechanism can take the general flexible form of a database predicate over the unified space, so that the user can query on any subset of the unified space (problem or solution variables). In addition, we show that derived attributes representing constraint conditions can also be added to the predicate. By this means retrieved case can approximate solutions under constraints, capable of further adaptation, for example using CSP methods [Marling *et al*., 2002; Purvis and Pu, 1996; Faltings, 1996].

In order to generalise the CBR method to the unified space, we propose the definition of nearest neighbour to a region (really just any subset of the space). Query predicates on the space correspond to such regions, and nearest neighbours in the case base are defined as those nearest to the predicated region. By nearest here, we take distance to the nearest point in the region. With this definition, we show that the standard CBR nearest neighbour method is a special case.

In this section we have introduced the main idea of the method. In the next section we propose the theoretical basis for the method. In Section 3.2 we show that the standard nearest neighbour method is a special case of the unified general method. In Section 4 we give an illustrative example of the method as applied to a simple "inverse" problem. In Section 5 we show how it can deal flexibly with "mixed" target constraints over both problem and solution spaces. In Section 7 we give an outline of a real practical example of a design assistant to the pneumatic conveyor.

2. GENERALISATION OF NEAREST NEIGHBOUR RETRIEVAL OVER THE N-DIMENSIONAL UNIFIED SPACE

2.1 Notation

In this section, we present the basis for retrieval of nearest neighbours to a constrained region over a unified n-dimensional space. We take the space to be formed from n one-dimensional domains: D_i , $i = 1, \ldots, n$, each with a distance function defined. If a and $b \in D_i$, we take $|a-b|$ to be the distance between elements a and b. We let D denote the n-dimensional space $D_1 \otimes D_2 \otimes D_3 \otimes \ldots D_n$, and let $(z_1, z_2, \ldots, z_n)$ represent a point in D, where $z_1 \in D_1$, $z_2 \in D_2, \ldots,$ and etc.

If $a, b \in D$ we assume $d(a,b)$ represents a distance function defined over D, such that

$$d(a,b) > 0, \; \forall \; a, b \qquad \text{E(3-1)}$$

$$d(a,a) = 0, \; \forall \; a \qquad \text{E(3-2)}$$

$$d(a,b) = d(b,a), \; \forall \; a, b \qquad \text{E(3-3)}$$

$$d(a,b) + d(b,c) \geq d(a,c), \; \forall \; a, b, c \qquad \text{E(3-4)}$$

NB. The existence of a metric d on the unified space which satisfies E(3-4) implies that the usual similarity assumption [Watson, 1997; Kolodner, 1993] holds true.

Normally we take d to be the pseudo-Euclidean metric:

$$d(a,b) = [(w_1 \,|\, a_1 - b_1 \,|^2 + w_2 \,|\, a_2 - b_2 \,|^2 + \ldots + w_n \,|\, a_n - b_n \,|^2) / \textstyle\sum w_{i=1 \ldots n}]^{1/2}$$

In many cases a user needs to specify a region R and we need to find its nearest neighbours within a given case base $C \subset D$. We first extend the definition of metric to include the distance between a point and a region. Let $R \subset D$ be a region in the n-dimensional space. We define:

$$d(a,R) = \min_{r \in R} d(a,r) \qquad \text{E(3-5)}$$

Now we use the definition E(3-1), applied to cases in C. For a given region R, we define the nearest neighbour as

$$Arg\min_{c \in C} d(c,R) \qquad \text{E(3-6)}$$

(Notice there is a symmetry here between R and C. Combining E(3-5) and E(3-6) we get:

$$Argmin_{c \in C} \; \min_{r \in R} d(a,r) \qquad \text{E(3-7)}$$

2.2 The Standard CBR Model as a Special Case

The above analysis treats all n dimensions of the unified space equally. However, in standard case base analysis, it is usual to divide the space into two parts: the problem space, which is completely known and specified as a target ($x_1, x_2, \ldots, x_k$), and the solution space ($y_1, y_2, \ldots, y_m$) which is unknown. To apply the general theory of the previous section, we consider the n dimensional space ($z_1, z_2, \ldots, z_n$) where $n = k+m$ and

$$z_1 = x_1, \ldots, z_k = x_k,\ z_{k+1} = y_1, \ldots,\ z_n = y_m.$$

The user specifies the target region R by the k constraints (see Fig. 1):

$$r \in R \text{ iff } r_1 = x_1^{target}, r_2 = x_2^{target}, \ldots, r_k = x_k^{target}.$$

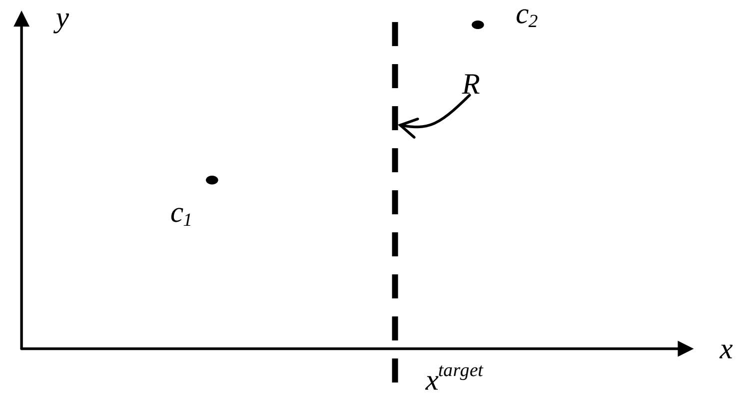

Figure 1. Visualisation of a target case, x^{target} as a constrained region *R*

If we take $d()$ as the pseudo-Euclidean metric, then

$$d(c,r)=[(w_1 |c_1 - r_1|^2 + w_2 |c_2 - r_2|^2 + \ldots + w_n |c_n - r_n|^2) / \sum w_{i=1\ldots n}]^{1/2}$$

For $i = k+1 \ldots n$, there is no constraint on r_i, so we can chose $r_i = c_i$, whereupon $|c_i - r_i| = 0$ for a minimum $d(c,r)$. For $i = 1\ldots k$, $r_i = x_i^{target}$. Hence E(3-7) gives

$$Argmin_{c \in C} [(w_1 |c_1 - x_1^{target}|^2 + \ldots + w_k |c_k - x_k^{target}|^2) / \sum w_{i=1\ldots n}]^{½}$$

which is the standard nearest neighbour formula, relying on a distance metric defined only on the problem space.

Hence we see that the generalised nearest neighbour retrieval reduces to the standard retrieval method when a target is available. The target is specified by means of constraints on the n-dimensional space and nearest neighbours are determined for this region. This justifies examining the method further. Now, it remains to investigate how this method behaves with other constraints.

3. THE PROJECTILE PROBLEM: THE INVERSE PROBLEM

The above analysis is obviously symmetric between x and y, i.e., problem and solution domain have no significance in this generalised approach. If the user specifies the target region R by the m constraints:

$$r \in R \text{ iff } r_1 = y_1^{target}, r_2 = y_2^{target}, \ldots, r_m = y_m^{target}.$$

The problem reduces to the nearest neighbour problem with a metric defined over the y-domain.

To illustrate how the method works, we choose a simple projectile problem. In this problem, the projectile model simulates the flight of a cannonball shot over flat ground in a given time interval, T (see Fig. 2). From numerical modelling point of view, the direct problem consists of two inputs; v = velocity, θ = angle of gun and the model calculates the output: trajectory of the shot. A trajectory is represented by a set of points, i.e., (x,y) where :

$$x = v\, t\, Cos\, \theta$$

$$y = v\, t\, Sin\, \theta - 5t^2$$

$$\text{and } \; t \in [0,T]\, .$$

In this representation, v and θ are the problem space, and (x,y) the solution space.

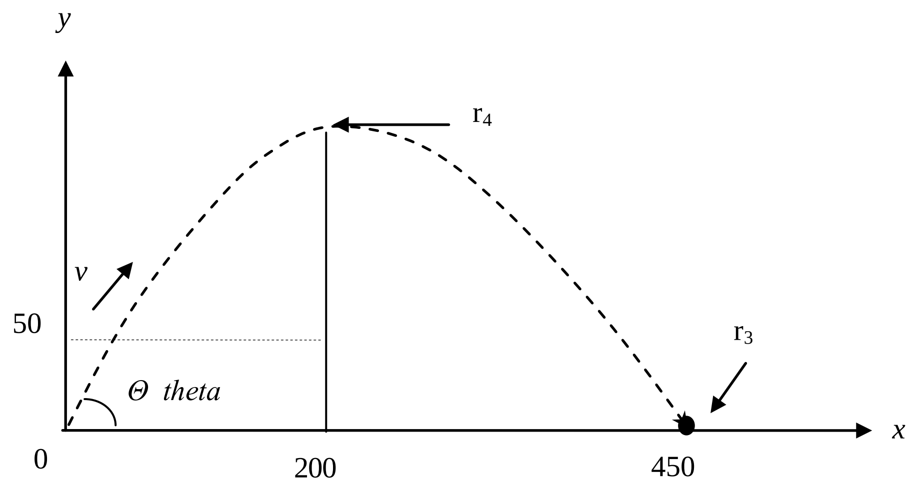

Figure 2. Visualisation of the trajectory of a cannonball shot over flat ground

To illustrate the inverse problem, we take an example where a gunner tries to hit the target at x=450m on the level ground. The problem is this: we want to decide on the right angle of gun, θ and the initial velocity, v. Here we assume that there are various guns, each with known muzzle velocity. We have no means to approach such an inverse problem with standard CBR models where problem space and solution space are treated as separate since we do not know the "inputs" in the problem space, only knowing that the "output" = 450m. Hence we can take R to be the space of points $r(r_1, r_2, r_3)$ where r_1 = theta, $r_2 = v$, r_3 = the x value of the point where the cannon ball hits the level ground. R is the region subject to the constraint: $r_3 = 450m$.

In the generalised approach, each case in the case base is represented by $c(c_1, c_2, c_3)$, and the metric is:

$$d(c,r) = [(w_1\,|c_1 - r_1|^2 + \; w_2\,|c_2 - r_2|^2 + \; w_3\,|c_3 - r_3|^2))/ \textstyle\sum w_{i=1...n}\,]^{\,1/2}.$$

As there are no constraints on r_1 and r_2 , *the minimum* $d(c,r)$ *will occur when* $r_1 = c_1$, so that $|c_1\text{-}r_1| = 0$.

Similarly, $|c_2\text{-}r_2| = 0$. Hence E(3-7) reduces to:

$$Arg\ min_{\,c\, \in C}\;\; [\, w_3 \mid c_3 - 450\, |^2 / \, w_3]^{\,1/2} = Arg\ min_{\,c\, \in C}\;\; |\, c_3 - 450\, |$$

and we are searching for cases with c_3 near to $450m$. Table 1 shows example of cases, i.e., $c(c_1, c_2, c_3)$, generated in case base with theta $\in (0\, , \pi/2]$ and velocity $\in [70\, , 100]$:

Case ID	c_1	c_2 *(m/s)*	c_3 *(m)*	*d (c,R)*
1	0.2	100	389.4183	60.5817
2	1.2	80	432.2964	17.7036
3	1.0	70	445.5557	4443
4	0.6	70	456.6992	6.6992
5	0.4	80	459.1079	9.1079
6	0.8	70	489.7911	39.7911
7	1.2	90	547.1252	97.1252

Table 1. Example cases presented in case base and the distance calculated for between each case and the target region *R* – The Inverse Problem.

With this approach, the inverse problem can be resolved. The retrieved cases for *k*=3 nearest neighbours are:
Case 3: (1, 70, 445.6), giving a near solution with muzzle velocity 70m/s and angle 1^c
Case 4: (0.6, 70, 456.7), and
Case 5: (0.4, 80, 459.1).

As case 3 gives minimum $d(c,R) = 44$, according to E(3-7) , the best match to the target region *R* is case 3.
The above shows how the generalised approach can be used to handle an inverse problem, however, there are times when users may need to specify a problem over the input space and the output space. Section 5 investigates how the method could be used to solve such constraint problems.

4. THE PROJECTILE PROBLEM: THE CONSTRAINT PROBLEM

The fact that standard CBR models restrict query only on the problem space may be a problem when users want to define a constraint over both the problem space and the solution space to refine the search. Comparatively, the generalized approach allows users to define constraints on both spaces.

4.1 Constraint Problem I

In this section, we show an example of "mixed" target constraints problem. For example, the gunner who tries to hit the target at x = 450m on the level ground may want to restrict the choice of gun with initial velocity = 80m/s. The target region *R* now involves constraints defined over the problem space (i.e., input) and the solution space (i.e., outputs): $r_2 = 80$ *m/s* , $r_3 = 450m$.

The metric is again:

$$d(c,r) = [(w_1 \mid c_1 - r_1|^2 + w_2 \mid c_2 - r_2|^2 + w_3 \mid c_3 - r_3|^2))/ \sum w_{i=1 \ldots n}]^{1/2}.$$

As there are no constraints on r_1, the minimum $d(c,r)$ will occur when $r_1 = c_1$, so that $|c_1 - r_1| = 0$. Hence following E(3-7) reduces to

$$Argmin_{\, c \in C} [(w_1 |c_1 - r_1|^2 + w_2 |c_2 - 80|^2 + w_3 |c_3 - 450|^2)) / \sum w_{i=1 \ldots n}]^{1/2}$$

$$= Argmin_{c \in C} [(w_2 | c_2 - 80 |^2 + w_3 | c_3 - 450 |^2) / \sum w_{i=1...n}]^{½}$$

Hence we are searching for cases with c_2 near to *80m/s* and c_3 near to *450m*. In this problem we take equal weights for w_2 and w_3 assuming that r_2 and r_3 are equally important. Table 2 shows example of cases, i.e., $c(c_1,c_2,c_3)$, generated in case base with theta $\in (0, \pi/2]$ and velocity $\in [70,100]$:

Case ID	c_1	c_2 *(m/s)*	c_3 *(m)*	*d (c,R)*
1	0.2	100	389.4183	45.1118
2	1.2	80	432.2964	12.5183
3	1.0	70	445.5557	7.7380
4	0.6	70	456.6992	8.5111
5	0.4	80	459.1079	6.4403
6	0.8	70	489.7911	29.0115
7	1.2	90	547.1252	69.0409

Table 2. Example cases presented in case base and the distance calculated for between each case and the target region *R* – Constraint Problem I.

Using the general method, the “mixed” target constraints problem can be resolved. The retrieved cases for *k*=3 nearest neighbours are:
Case 5: (0.4, 80, 459.1), giving a near solution with muzzle velocity 80m/s and angle 0.4^c
Case 3: (1, 70, 445.6), and
Case 4: (0.6, 70, 456.7).
As case 5 gives minimum $d(c,R) = 6.44$, according to E(3-7) , the best match to the target region R is case 5.

5. Constraint Problem II

In Section 6, we continue the discussion of handling constraints using the generalised approach by adding a new constraint to the example discussed in Section 5.1 . In this example, we add an extra constraint in the form of an obstacle at distance 200m that the cannon ball must clear. We suppose that the gunner tries to hit the target at x = 450m on level ground with a choice of gun with initial velocity = 80m/s and clear the obstacle with height 50m. To handle this constraint, we augment the space to include a derived dimension, r_4 = the height of the cannon ball at the obstacle, when x = 200m.
By adding a new constraint, the constrained region R is defined by : $r_2 = 80$ *m/s* , $r_3 = 450m$, $r_4 > 50m$, where r_4 = the height of the cannon ball at the obstacle, when x = 200m. We take the distance metric on this dimension to be:

$$d(a_4,b_4) = \begin{cases} |a_4-50|, & a_4 <= 50, b_4 > 50 \\ 0, & \text{otherwise} \end{cases}$$

With the generalised approach we reformulate the problem so that the metric bears a penalty term $w_4\, d^2(c_4, r_4)$ for cases outside the constraint region:

$$d(c,r) = [(w_1 | c_1 - r_1|^2 + w_2 | c_2 - r_2|^2 + w_3 | c_3 - r_3|^2 + w_4\, d^2 (c_4, r_4)) / \sum w_{i=1...n}]^{1/2}$$

Since no constraint is defined on r_1, following E(3-7) the metric reduces to

$$Argmin_{\, c \in C} \; [\, (w_1 \mid c_1 - I_1 \mid^2 + \; w_2 \mid c_2 - 80 \mid^2 + \; w_3 \mid c_3 - 450 \mid^2 + \; w_4\, d^2\, (\, c_4, r_4)\,)\, / \sum w_{i=1 \ldots n} \;\;]^{½}$$

$$= \; Argmin_{\, c \in C} \; [(w_2 \mid c_2 - 80 \mid^2 + \; w_3 \mid c_3 - 450 \mid^2 + \; w_4\, d^2\, (\, c_4, r_4))\, / \sum w_{i=1 \ldots n} \;]^{\,1/2}$$

Hence we are searching for cases with c_2 near to *80m/s* , c_3 near to *450m* and c_4 greater than *50m*.
In this problem, we take the weights for the constraints to be high, since we definitely want to use a gun with muzzle velocity 80m/s, and we cannot afford to hit the obstacle. In this example, we took $w_2 = 1$, $w_3 = 10$, $w_4 = 10$, which is high for constrains r_3 and r Table 4-3 shows example of cases, i.e., $c\,(\, c_1 ,\, c_2 ,\, c_3 ,\, c_4\,)$, generated in case base with theta ∈ $(0 , \pi/2]$ and velocity ∈ $[70,100]$:

Case ID	c_1	c_2*(m/s)*	c_3 *(m)*	c_4 *(m)*	*d (c,R)*
1	0.2	100	389.418	19.720	46.940
2	1.2	80	432.296	276.432	12.217
3	1.0	70	445.556	171.665	3.7638
4	0.6	70	456.699	76.907	5.112
5	0.4	80	459.108	47.723	6.479
6	0.8	70	489.791	121.840	27.545
7	1.2	90	547.125	326.382	67.058

Table 3. Example cases presented in case base and the distance calculated for between each case and the target region *R* – Constraint Problem II.

Using the general method, again, the "mixed" target constraints problem can be resolved. The retrieved cases for *k*=3 nearest neighbours are:
Case 3: (1 , 70, 445.6 , 171.7),
giving a near solution with muzzle velocity 70m/s and angle 1^c,
Case 4: (0.6 , 70, 456.7 , 76.9), and
Case 5: (0.4 , 80, 459.1 , 47.7).
As case 3 gives minimum $d(c,R) = 3.76$, according to E(3-7) , the best match to the target region *R* is case 3. In this case, the trajectory will clear the obstacle at a height of 171.67 m.

6. APPLYING THE METHOD TO THE PNEUMATIC CONVEYOR PROBLEM

To examine how the method could be useful for a real problem, we present an example of used in a CBR model for a pneumatic conveyor problem [Chapelle *et al.*, 2003].

Pneumatic conveying is an important transportation technology in conveying solid bulks in industry. Attrition of powders and granules during pneumatic conveying is a problem that has existed for a long time. One of the major industry concerns is to investigate how parameters such as air velocity, loading ratio, the angle of the bend and etc. affect degradation. Such knowledge is of great use in the design of conveyors.

Generally [Kalman, 2000], parameters affecting the attrition rate can be divided into three categories:

- The particle strength: particle material, size and shape.
- The operation parameters: particle velocity and particle concentration – loading ratio.
- The pipeline and bend structure: radius of curvature, construction material, type of bend, number of bends.

In most cases, pneumatic conveyor engineers are more concerned with the set of input parameters (i.e., as mentioned above) that will produce a desirable size distribution of particles (see Fig. 3).
From a designer's point of view the problem is this: we want to decide on the right angle of bend, diameter of pipe, and air velocity for a given particulate, e.g., sugar or tea. We have a number of bend angles and three pipe diameters available. The sugar must not degrade so that there is too much dust formed (i.e., very small particles). There may be only low power fans available sometimes, so the air velocity may be constrained. What is the best setup to use?

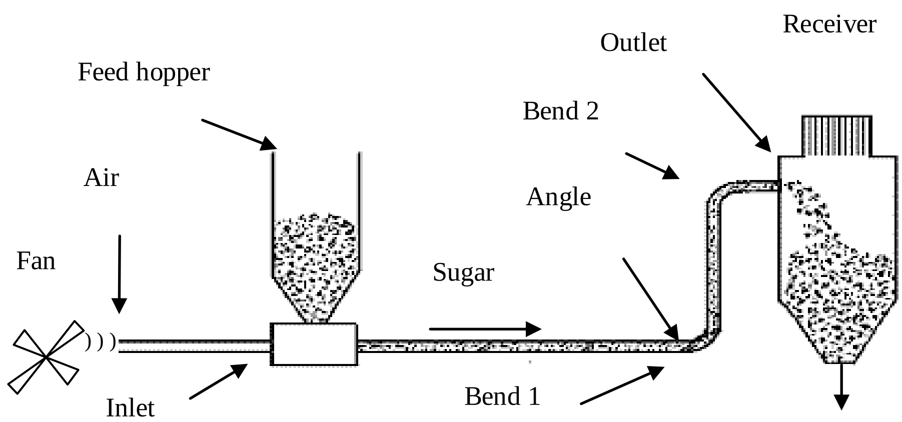

Figure 3. The schematic diagram of a sample pneumatic conveyor

To solve these inverse or constraint problems, a CBR model is built to help engineers. The model uses the generalised nearest neighbour method to handle constraints in order to retrieve the best match to a target problem.

In the problem of pneumatic conveyor, the performance of the CBR model explained in this paper is measured against the human expert [Chapelle *et al.*, 2003] in solving inverse problems and constraint problems in the pneumatic conveyor domain. Results [Knight and Woon, 2004] show that the CBR model is on average 5 times faster than the human expert in using the numerical model directly to search for a solution. The CBR model provides a good solution in terms of efficiency and effectiveness. It is shown that the CBR model using this approach can help engineers to solve inverse and constraint problems in the pneumatic conveyor domain.

7. CONCLUDING REMARKS

In this paper we have presented a generalised version of CBR model based on a unified (problem: solution) space. The nearest neighbour retrieval method has been re-cast as nearest neighbour to a constrained region. The region may be defined by means of a predicate defined on the unified space. This allows the user greater flexibility in specifying any query over the unified space.

It is also shown that the user can add constraints to the query by adding derived attributes to the predicate. The retrieval then searches for the nearest neighbours in the enhanced space (unified space + constraint attributes). Solutions will be retrieved, which satisfy or are close to the constrained region. The paper has presented a mathematical basis for these proposals, and shown that the standard CBR method is a special case. It illustrates the flexibility of query mode by means of an inverse problem, and illustrates the addition of constraints to a projectile problem. The details of a practical project which has utilised this approach have been presented as the problem of a pneumatic conveyor design. Future work is planned on the use of the method in conjunction with

CSP in conjunction with the Generalised Shepard Nearest Neighbour interpolation method [Knight and Woon, 2003; Woon and Knight, 2004].

References

[Chapelle *et al*., 2003] Chapelle, P., Christakis, N., Abou-Chakra, H., Bridle, I., Bradley, M.S.A., Patel, M., & Cross, M. (2003) Computational Model for Prediction of Particle Degradation during Dilute Phase Pneumatic Conveying: Modelling of Dilute Phase Pneumatic Conveying, Advanced Powder Technology

[Faltings, 1996] Faltings, B. (1996) Working Group in Model-Based Design and Reasoning. Part II: Design. AI Communications, 9 (2) pp.65-73

[Kalman, 2000] Kalman, H. (2000) Attrition of Powders and Granules at Various Bends during Pneumatic Conveying. Powder Technology, 112 pp. 244-250

[Knight and Woon, 2004] Knight, B. & Woon, F. L. (2004) Case Based Adaptation Using Interpolation over Nominal Values. In: Proceedings of AI-2004, The 24th Specialist Group on Artificial Intelligence (SGAI) International Conference on Innovative Techniques and Applications of Artificial Intelligence, Research and Development in Intelligent Systems XXI, Cambridge, UK, December, London, Springer-Verlag, pp. 73-86

[Marling *et al*., 2002] Marling, C., Sqalli, M., Rissland, E., Hector, M. & Aha, D. (2002) Case-Based Reasoning Integrations. AI Magazine. USA, AAAI Press

[Purvis and Pu, 1996] Purvis, L. & Pu, P. (1996) An Approach to Case Combination, Workshop on Adaptation in Case-Based Reasoning. In: 12th European Conference on Artificial Intelligence, 11-16 August, Budapest, Hungary.

[Woon *et al*., 2003b] Woon, F. L., Knight, B. & Petridis, M. (2003) Case-Based Reasoning as a Tool to Improve the Usability of Numerical Models. Workshop Proc. of 5th International Conference on Case-Based Reasoning, ICCBR-03, Trondheim, Norway, June 23-26, NTNU Department of Computer and Information Science technical report 4/2004, pp.200-209

[Woon *et al*., 2003c] Woon, F. L., Knight, B., Petridis, M., Chapelle, P. & Patel, M. (2003) Enhancing the Usability of Numerical Models with Case-Based Reasoning. In: Lees, B. (ed.) Proc. of 8th UK Workshop on Case-Based Reasoning, AI-03, Cambridge, UK, December 15, pp.38-46