The Property of Different Granule and Granular Methods Based on Quotient Space

Yan-ping Zhang, Ling Zhang and Chenchu Xu

Abstract Nowadays, we have entered the era of big data, and we have to deal with complex systems and massive data frequently. Facing complicated objects, how to describe or present objects is the base to solve questions frequently. So we suppose that a problem solving space, or a problem space for short, is described by a triplet (X, f, Γ), and assume that X is a domain, R is an equivalence relation on X, Γ is a topology of X, $[X]$ is a quotient set under R. Regarding $[X]$ as a new domain, we have a new world to analyse and to research this object, consequently we describe or present a question into different granule worlds, these granular worlds are called the quotient space. Further we are able to predigest and solve a question, i.e. we apply quotient space and granulate to represent an object. Comparing rough set and decision-making tree, the quotient space has the stronger representation. Not only it can represent vectors of the problem domain, different structures between vectors, but also it can define different attribute functions and operations etc. In this paper, we discuss the method how to represent and to partition an object in granular worlds, and educe the relationship of different granular worlds and confirm the degree of granule. We will prove three important theorems of different granules, i.e. to preserve false property theorem and to preserve true property theorem. To solve a problem in different granular worlds, the process procedure of quotient approximate will be applied. We also supply an example of solving problem by different granule worlds—the shortest path of a complex network. The example indicates that to describe or present a complicated object is equal to construct quotient space. In quotient set $[X]$, the complexity to solve a problem is lower than X. We have a new solution method to analysis a big data based on the quotient space theory.

Y. Zhang (✉) · L. Zhang
School of Computer Science and Technology, Anhui University, Hefei 230039, China
e-mail: zhangyp2@gmail.com

C. Xu
Anhui Provincial Hospital, Hefei 230001, China
e-mail: 443779952@qq.com

W. Pedrycz and S.-M. Chen (eds.), *Information Granularity, Big Data, and Computational Intelligence*, Studies in Big Data 8,
DOI: 10.1007/978-3-319-08254-7_8,

Keywords Quotient space · Granular computing · Equivalence relation · The shortest path

1 Introduction

When we confront complex problems which are hard to handle them accurately, it's not usual to pursue the optimal solution in either systematic or precise way. On the contrary, we reach the limited and reasonable destination step by step, in one way or another, it means we achieve the so-called satisfactory solution. Thanks to the multi-granularity analysis which is sketchy, from coarse to fine and more and more accurate, we successfully avoid the difficulties on the computational complexity. Just in this way, a lot of nonpolynomial questions are smoothly solved.

It is just said by Zhang in [1, 2] "One of the basic characteristics in human problem solving is the ability to conceptualize the world at different granularities and translate from one abstraction level to the others easily, i.e. deal with then hierarchically." Because of the differences of the point of observing an object and the object's further information, a complicated object can be briefed some points that reserve the important characteristics and performances according to the demand to analyse and solve a problem. These points are the representation of different granule worlds.

Existing studies of granular computing typically concentrate on concrete models and computational methods in particular contexts. They unfortunately only reflect specific aspects of granular computing. In fact, there does not exist a formal, precise, commonly agreed, and uncontroversial definition of what is granular computing, nor there is a unified model. Consequently, the potential applicability and usefulness of granular computing are not well perceived and appreciated [3]. Many methods and models of granular computing have been proposed and studied [4–9]. The results enhance our understanding of granular computing. Granular computing comes with a number of interesting pursuits [10–12]. The idea of information granulation offers immediate advantages. It provides tangible benefits in fuzzy modeling by supporting meaningful ways of striking a sound balance between interpretability and accuracy of fuzzy models [13–16], they offer some ways of assessing the performance of the model formed in this way.

The granule world that we define is different from the information granule (IG) that Pawlak proposes. The information granule what is said is a kind of reflection of limited abilities that people deal with and store information, i.e. when facing a lot of complicated information and having the limited abilities, people need partition the information into some simple information blocks according to each characteristic and performance in order to deal with easily. The information block is thought a granule [17–20]. Because information granules are partitioned according to equivalence relation, this only changes the granule of domain of problems and attributed relationship, and the space structure does not been changed. The representation of different granule worlds—quotient space in this paper changes granules not only the domain and attributed relationship but also the space structure of a problem.

In 1990, Bo Zhang and Ling Zhang firstly proposed a new theory, Quotient Space Theory (briefly QST) [1], which was pay high attention to by domestic and overseas scholars. In 1992, the monograph on QST Theory and Applications of Problem Solving [2] was published. In clear–cut classification, we use equivalence relation for establishing our model. A natural question is whether fuzzy equivalence relation can be used for constructing fuzzy classification model. So we have done some research on fuzzy quotient space [21–23]. Recently, we try to use QST to analysis complex networks and some dynamic information [24–27].

When QST contrast with Zadeh's granule computing [28–30], it will transform the original quotient space to fuzzy quotient space with the aid of fuzzy relation of equivalence. Owing to the condition, we think both are similar.

In this paper, we discuss the method how to represent and to partition an object in granular worlds, and educe the relationship of different granular worlds and confirm the degree of granule. We will prove two important theorems of different granules, i.e. to preserve false property theorem and to preserve true property theorem. To solve a problem in different granular worlds, the process procedure of quotient approximate will be applied. We also supply an example of solving problem by different granule worlds—the shortest path of a complex network. The example indicates that to describe or present a complicated object is equal to construct quotient space. In quotient set $[X]$, the complexity to solve a problem is lower than X.

2 Quotient Space

2.1 Basic Definition

A problem solving space, or a problem space for short, is described by a triplet (X, f, Γ). X denotes the problem domain, $f(\cdot)$ indicates the attributes of domain X or is denoted by a function f: $X \to Y$, Γ is the structure of domain X, i.e. the relationship among elements in X. To analyse and solve the triplet (X, f, Γ) of a problem implies analysis and investigation of X, f and Γ.

Assume that X is a domain, R is an equivalence relation on X, $[X]$ is a quotient set under R. Regarding $[X]$ as a new domain, we have a new world which is coarser than X. So a quotient space is a new world that an equivalence relation is thought a new element and coarser than X.

Definition 2.1 A quotient space is the object representation using the quotient set of mathematics to describe or present different granule worlds, i.e. is the method using the quotient set as the mathematic model of different granule worlds.

Problem representations between different granularity sizes correspond to different equivalence relation R or different partitions. So how to partition is the method to construct different granule worlds. We can classify $X = f^{-1}(Y)$ by using the result Y, or classify X directly. In detail there are several methods which supply in next paragraph.

(1) Attribute-based method, namely the same attributions or similar elements are classified.
(2) Projection-based method, consider f is multi-dimensional. Let its n attribute components be $f_1, f_2, \ldots, f_i, f_{i+1}, f_{i+2}, \ldots, f_n$. X is classified with respect to $f_{i+1}, f_{i+2}, \ldots, f_n$ values, while ignoring their attribute components $f_1, f_2, \ldots, f_i$.
(3) Function-based method, a set X of elements is partitioned according to their functions or structures.
(4) Constraint-based method, given n constraints $C_1, C_2, \ldots, C_n$ and a domain X, we may partition X according to C_i, $i = 1, 2, \ldots, n$.

In some cases, some $x \in X$ may belong to more than one class. That is, the classification has overlapped elements or the contour of classes are blurred. We can introduce fuzzy logic for these cases.

Generally, we treat a problem under various grain sizes. Thus, it is necessary to establish the relationship between the worlds at different granularities.

In the book [1, 2] Zhang has discussed the relation between X and $[X]$ and showed that the domains of different granularities are a complete semi-order lattice. But for a problem space (X, f, Γ), structure Γ is very important. When a domain X is discomposed, its structure will change as well. Generally, the coarser the granularities are, the simpler the structure is, however, are there changes of the structure after predigested?

(X, Γ) is a topologic space and Γ is a topology on X. Assume that R is an equivalence relation on X. From R, we have a quotient set $[X]$. A topology $[\Gamma]$ on $[X]$ induced from Γ is called a quotient topology, and $([X], [\Gamma])$ is a quotient topologic space. From topology, it is known that some properties of topologic space (X, Γ) can be observed from its quotient space $([X], [\Gamma])$. We have

Proposition 2.1 *Assume that* p: $(X, \Gamma) \to ([X], [\Gamma])$ *is a continuous mapping. If* $A \subset X$ *is a connected set on* X, *then* $p(A)$ *is connected set on* $[X]$.

Proposition 2.1 shows that if there is a solution path (connected) in the original domain X, then there exists a solution path in its proper coarse-grained domain $[X]$. Conversely, in the coarse-grained domain, if there does not exist a solution path, there is no solution in the original domain. These properties show that a quotient space has the characteristic of reserving false.

(X, Γ) is a semi-order space or a pseudo semi-order space. In order to establish some relations between semi-order spaces at different grain sizes, we expect to induce a structure $[\Gamma]$ of $[X]$ form Γ of X such that $([X], [\Gamma])$ is also a semi-order space and for all $x, y \in X$, if $x < y$ then $[x] < [y]$. Namely, it is desired that the order relation is preserved invariant in grain size, or order-preserving for short. We can follow the next steps:

(1) transform (X, Γ) into some sort of topologic space.
(2) construct a quotient topologic space $([X], [\Gamma])$ form (X, Γ).
(3) induce a semi-order from $([X], [\Gamma])$ such that the original order relations are preserved in the space.

Proposition 2.2 *Suppose that R is compatible with* Γ. *If* $x, y \in (X, \Gamma)$ *and* $x < y$, *then* $[x] < [y]$, *where* $[x], [y] \in (X, \Gamma)$.

Proof Assume that x is littler than y, define that a is equal to $[x]$.

Assume that $u(a)$ is any opening domain of a on $[X]$, because p: $X \to [X]$ is continuous and $p^{-1}(u(a))$ is not closing on X, thus x is a domain.

Because of $x < y \Rightarrow y \in p^{-1}(u(a)) \Rightarrow p(y) \in u(a) \Rightarrow [y] \in u(a)$, namely $[x] < [y]$. □

Proposition 2.2 indicates that the quotient semi-order space constructed by the preceding approach has order-preserving.

If X is very complicated, we can introduce an equivalence relation and transform X into $[X]$. If R is compatible with Γ, then induces a quotient semi-order $[\Gamma]$ on $[X]$. Thus the old question from x to y is transformed into the new question from $[x]$ to $[y]$. Because R is compatible, then p: $[X, \Gamma] \to ([X], [\Gamma])$ is order preserving. Namely suppose that $[X]$ is a coarse-grained level of X. If there is no solution on some regions of $[X]$, from Proposition 2.2, it is known that there is no solution on the corresponding regions of X as well. Based on the principle, the searching range will be narrowly by pruning off those areas. Since the coarse-grained world usually is simpler than the original one, the searching efficiency will be improved.

Generally, it is not necessary that all characteristics on (X, Γ) are completely mapped onto $([X], [\Gamma])$. This means that in the coarse-grained world some information might be missed due to the abstraction. If the missing characteristics are not interested, that does not matter very much. But the main ones must be preserved in $[X]$.

2.2 To Select Proper Grain-Size

Facing different researching goals, there are different quotient sets $[X]$ from the same object (X, f, Γ), and there are different quotient structures $[\Gamma]$ on the same quotient set. Thus how to select proper grain-size is the key to construct reasoning $[X]$, $[\Gamma]$.

In really, the partition is dynamic, namely we partition a X and investigate and research the object and extract some properties on the grain-size firstly, then partition the X again, and so on till the problem is solved. Generally selection and adjustment of grain-size is relation with main domanial acknowledge of the problem. We can select and adjust grain-size by mergence and decomposition.

By merging, we have a new equivalence relation $\underline{R}$ such that (1) $\underline{R} < R$, (2) $\underline{R}$ is compatible, and (3) $\underline{R}$ is maximum. That is, $\underline{R}$ may change R such that the R is compatible and is the coarsest under the supplied condition, namely the number of partition is the least.

By decomposing, we have a new equivalence relation $\overline{R}$ such that (1) $R < \overline{R}$, (2) $\overline{R}$ is compatible, (3) $\overline{R}$ is minimum. That is, $\overline{R}$ may change R such that the R is compatible and is the finest under the supplied condition, namely the number of partition is the biggest.

Thus if R is incompatible with Γ, we can adjust R by merging or decomposing in order that R is compatible, and the compatible result is uniform in condition of maximum or minimum.

The approach we offer for constructing quotient semi-order is the following. First, aright-order topology Γ_R is induced from semi-order Γ. Second, a quotient topology $[\Gamma_R]$ on $[X]$ is induced from Γ_R. Third, a semi-order on $[X]$ is induced from $[\Gamma_R]$. And if R is incompatible with Γ, we can adjust R by merging or decomposing in order that R is compatible.

3 Property Preserving Ability

3.1 Falsity Preserving Principle

We have defined the relation between the domains $[X]$ and X. For a problem space (X, f, T), structure T is very important. When a domain X is decomposed, its structure will change as well. Generally, it is simplified. The main point is whether some properties (or attributes) in X that we are interested in are still preserved after the simplification.

Proposition 3.1 *Assume that R is compatible. If* $x, y \in [x]$ and $x < y$, then interval $[x, y] = \{z | x < z < y, z \in X\} \subset [x]$.

Proof From $x < z < y$ and Proposition 2.2, we have $[x] < [z] < [y]$. Since $[x] = [y] \Rightarrow [x] < [z], [z] < [x]$, from the compatibility of R, we have $[z] = [x] \Rightarrow z \in [x] \Rightarrow [x, y] \subset [x]$. □

Definition 3.1 (X, T) is a semi-order set. $A \subset (X, T)$ is connected $\Leftrightarrow \forall x, y \in$ A, $\exists x = z_1, z_2, \ldots, z_n = y$, such that z_i and z_{i+1}, $i = 1, 2, \ldots, n-1$, are compatible.

Definition 3.2 (X, T) is a semi-order set. $A \subset (X, T)$ is a semi-order closure $\Leftrightarrow$ if $x, y \in A$, and $x < y$, then interval $[x, y] \subset A$.

Corollary 3.1 *In assuming that R is compatible, each component of* $[X]$ *must consist of several semi-order closed, mutually incomparable, and connected sets.*

Note that sets A and B are mutually incomparable, if for $\forall x \in A, y \in B$, x and y are incomparable.

Proof [X] is divided into the union of several connected components, obviously, these components are mutually incomparable. We'll prove that each component is semi-order closed below.

Assume that A is a connected component of [X]. If A is not semi-order closed, then $\exists x_1, x_2 \in A$ and $y \notin A$ such that $x_1 < y < x_2$. Since R is compatible, $p : X \to [X]$ is order-preserving. We have $p(x_1) < p(y) < p(x_2) \Rightarrow [x] < [y] < [x] \Rightarrow [y] = [x]$.

That is, $y \in [x]$. Since $y \notin A$, y must belong to another connected component B of [X]. Thus, $x_1 \in A$ is comparable with $y \in B$. This contradicts with that components A and B are incomparable. □

Corollary 3.2 *If X is a totally ordered set and R is compatible, then each equivalence class of* [X] *must be an interval* $\langle x,y \rangle$, *where interval* $\langle x,y \rangle$ *denotes one of the following four intervals*: $[x, y], [x, y), (x, y], (x, y)$.

Especially, when $x = R^1$ (real number set), Corollary 3.2 still holds.

From the above corollaries, it's known that when partitioning a semi-order set with respect to R, only the corresponding equivalence classes satisfy some structure as shown in above corollaries so that R is compatible. In order to rationally partition a semi-order set, strong constraints have to be followed.

Proposition 3.2 (X, T) *is a semi-order set, then* $((X, T)_r)_s = (X, T)$.

Note: $((X, T)_r)_s$ *is a right semi-order set.*

From the previous discussion, it concludes that a quotient semi-order set ([X], [T]) can be induced from a semi-order set (X, T) so long as R is compatible. And ([X], [T]) has order-preserving ability.

When X is a finite semi-order set, it can be represented by a directed acyclic network G. And $x < y \Leftrightarrow$ there exists a directed path in G from x to y. When X is a finite set, X can be represented by a spatial network. We present a simple method for constructing a quotient (pseudo) semi-order on [X] below.

Given (X, T) and an equivalence relation R, we have a quotient set [X]. Define a relation "<" on [X] as $\forall a, b \in [X], \exists x_1 \in a, x_2 \in b, x_1 < x_2 \Rightarrow a < b$. Finding the transitive closure of relation "<", have a pseudo semi-order $[T]'$ and quotient space ([X], $[T]'$).

Proposition 3.3 $[T]' = [T]_r]_s$ *holds*, *where* $[T]'$ *as defined above*.

Using Proposition 3.3, when X is a finite set, the directed graph corresponding to (pseudo) semi-order on [X] can easily be defined as follows: $\forall a, b \in [X], a \to b \Leftrightarrow \exists x, y \in X, x \in a, y \in b, x < y$, where $a \to b$ means there exists a directed edge between a and b. The (pseudo) semi-order corresponding to the directed graph is just the quotient structure on [X].

3.2 Falsity (Truth) Preserving Principle

The order-preserving ability among different grain-size worlds has an extensive application. For example, the relation among elements of domain X is represented by some semi-order structure. A starting point $x \in X$ is regarded as a premise and a goal point $y \in X$ as a conclusion. Whether the directed path from point x to point y exists corresponds to whether conclusion y can be inferred from premise x. If X is complex, introducing a proper partition R to X, then we have $[X]$. A quotient (pseudo) semi-order $[T]_s$ on $[X]$ can be induced. Due to the following proposition, the original directed path finding from x to y on X is transformed into that from $[x]$ to $[y]$ on $[X]$.

Proposition 3.4 (X, T) *is a semi-order set. R is an equivalence relation on X. For* $x, y \in X$, *if there exists a directed path from x to y on* (X, T), *there also exists a directed path from* $[x]$ *to* $[y]$ *on* $[X]$.

The proposition shows that if the original problem (domain) in hand is too complex, by a proper partition, the original domain is transformed into a coarse one. If there does not exist a solution in the coarse world, then the original problem does not have a solution as well. Since the coarse world is generally simpler than the original one, the problem solving will be simplified.

Note that in Proposition 2.2, even R is incompatible, the order-preserving ability still holds.

From the previous discussion, it is know that an "inference" can be transformed into a spatial search from a premise to a conclusion, i.e., a path-search in a topologic space. And if an original problem (x, f, T) is too complex, then the problem can be transformed into its quotient space $([X], [f], [T])$ which generally simpler than the original one. The order-preserving and the falsity (truth) preserving ability that we will mention below clarify the main characteristics of the multi-granular world; which provide a theoretical foundation for multi-granular computing (inference).

Theorem 3.1 (Falsity Preserving Principle) *If a problem* $[A] \rightarrow [B]$ *on quotient space* $([X], [f], [T])$ *has no solution, then problem* $A \rightarrow B$ *on its original space* (X, f, T) *has no solution either. In other words, if* $A \rightarrow B$ *on* (X, f, T) *has a solution, then* $[A] \rightarrow [B]$ on $([X], [f], [T])$ *has a solution as well.*

Proof If problem $A \rightarrow B$ has a solution, then A and B belong to the same (path) connected set C of (X, f, T). Let $p : X \rightarrow [X]$ be a natural projection. Since p is continuous, $p(C)$ is (path) connected on $([X], [f], [T])$. $p(A) = [A]$ and $p(B) = [B]$ belong to the same (path) connected set of $([X], [f], [T])$. The problem $[A] \rightarrow [B]$ has a solution. □

Falsity preserving ability within a multi-granular world is unconditional but truth preserving is conditional.

Theorem 3.2 (Truth Preserving Principle I) *A problem* $[A] \to [B]$ *on* $([X], [f], [T])$ *has a solution, if for* $[x]$, $p^{-1}([x])$ *is a connected set on* X, *problem* $A \to B$ *on* (X, f, T) *has a solution.*

Proof Since problem $[A] \to [B]$ on $([X], [f], [T])$ has a solution, $[A]$ and $[B]$ belong to the same connected component C. Letting $D = p^{-1}(C)$, we prove that D is a connected on X.

Reduction to absurdity: Assume that D is partitioned into the union of mutually disjoint non-empty open close sets D_1 and D_2. For $\forall a \in C$, $p^{-1}(a)$ is connected on X, then $p^{-1}(a)$ only belongs to one of D_1 and D_2. $D_i, i = 1, 2$, composes of elements of $[X]$. There exist C_1, C_2 such that $D_1 = p^{-1}(C_1), D_2 = p^{-1}(C_2)$. Since $D_i, i = 1, 2$, are open close sets on X and p is a natural projection, C_1, C_2 are non-empty open close sets on $[X]$. And C_1 and C_2 are the partition of C, then C is non-connected. This is a contradiction. □

Theorem 3.3 (Truth Preserving Principle II) (X_1, f_1, T_1) *and* (X_2, f_2, T_2) *are two quotient spaces of* (X, f, T). $T_i, i = 1, 2$ *are semi-order.* (X_3, f_3, T_3) *is the supremum space of* (X_1, f_1, T_1) *and* (X_2, f_2, T_2). *If problems* $A_1 \to B_1$ *and* $A_2 \to B_2$ *have a solution on* (X_1, f_1, T_1) *and* (X_2, f_2, T_2), *respectively, then problem* $A_3 \to B_3$ *on* (X_3, f_3, T_3) *have a solution, where* $A_3 = A_1 \cap A_2$, $B_3 = B_1 \cap B_2$.

3.3 Computational Complexity Analysis

Using the falsity and truth preserving principle, the computational cost of the multi-granular computing (or inference) can greatly be reduced. For example, by choosing a proper quotient space and using falsity preserving principle, the part of the space without solution can be removed for further consideration so that the computing is accelerated. Similarly, by choosing a proper quotient space and using the truth preserving principle I, the problem solving on the original space can be simplified to that on its quotient space. In general, the size of the quotient space is much smaller than that of the original one so the computational cost is reduced. Concerning the truth preserving principle II, assume that n and m are the potentials of X_1 and X_2, respectively. The potential of X_3 is nm at most. Let $g(\cdot)$ be the computational complexity. When the problem is solved on X_3 directly, $g(\cdot) = g(nm)$. If using the truth preserving principle II, the problem is solved on X_1 and X_2, separately, the computational complexity is $g(n) + g(m)$. This is equivalent to reducing the complexity from $O(N)$ to $O(\ln N)$.

Note that T_3, a topology on (X_3, f_3, T_3), is not necessarily an induced quotient topology $[T_3]$ of X_3, generally $T_3 < [T_3]$. Namely, (X_3, f_3, T_3) is only an element of complete semi-order lattice $\boldsymbol{V}$ but $\boldsymbol{U}$.

4 The Hierarchical Quotient-Space Model of Complex Networks

We have already set up the model of quotient space, and found the Theorems 3.1–3.3 which indicated the change of the main characters during the procedure of granular computing.

In the era of big data, we are forced to confront complex big data sets. In order to analysis these data sets, we utilize multi granular spaces based on quotient spaces. During the process we may reduce the complexity of the data set and solve these big data sets.

Next we present a hierarchical quotient-space model that reduces the computational complexity. We discuss the model to solve the shortest path in a complex network. A complex network is represented by an undirected weighted graph (X, E), where X is a set of nodes, E is a set of edges, f: $E \to R+$, $f(e) \in [0, w]$ is the weight of edge e. Weight w indicates flux, bandwidth or traffic, etc., that is, the reciprocal of distance. An optimal path is the path that connects any pair of nodes with the maximal weight. Then, a shortest path is the path with minimal distance (the reciprocal of weight). Let a set of weights on edges be $\{w^1 > w^2 > \cdots > w^k\}$. In the following discussion, space (or graph) is denoted by (X, E), or simply by X.

Definition 4.1 Equivalence relation $R(w_i)$ is defined as

$$x \sim y \Leftrightarrow \exists x = x_1, x_1, \ldots, x_m = y, f(x_j, x_{j+}) \geq w_i, j =, \ldots, m-1, i = 1, \ldots k$$

Define $X_i = \{x_1^i, \ldots, x_{n_i}^i\}$, $i = 1, \ldots, k$ as a quotient space corresponding to $R(w_i)$. Let $X = X_0$, and x_i^0 be the element of X. Ranking the elements (nodes) of quotient space X_i, we have a space denoted by $X_i = \{x_1^i, \ldots, x_{n_i}^i\}$, $i = 1, \ldots, k$, as well. Obviously, $(X_0, X_1, \ldots, X_k)$ forms a sequence of hierarchical quotient spaces. Now, the elements in space X are represented by a hierarchical encoding as follows.

For $z \in X$, z is represented by a $(k+1)$-dimensional integral $z = (z_0, z_1, \ldots, z_k)$. Assume that $p_i : X \to X_i$ is a natural projection. If $p_i(z) = x_t^i$, let the ith coordinate of z be t, i.e., $z_t = t$. It means that if z belongs to the tth element of X_i, then $z_i = t$.

For space X, define a set of its edges as $E_0 : e = (x_j^0, x_t^0) \in E_0 \Leftrightarrow f(x_j^0, x_t^0) \geq w_1$. Simply, let edge $e(x_j^0, x_t^0) = (x_j^0, x_t^0)$. This way, we construct (X_0, E_0).

For space X_1, define a set of its edges as

$E_1 : (x_j^1, x_t^1) \in E_1 \Leftrightarrow \exists x_j^0, x_t^0 \in X, x_j^0 \in x_j^1, x_t^0 \in x_t^1, f(x_j^0, x_t^0) \geq w_2$. Edge $e(x_j^1, x_t^1)$ is represented by

$$e(x_j^1, x_t^1) = \{((x_j^0, p_1(x_j^0)), (x_t^0, p_1(x_t^0))) | \forall x_j^0, x_t^0 \in X, x_j^0 \in x_j^1, x_t^0 \in x_t^1, f(x_j^0, x_t^0) \geq w_2\}$$

Edge $e(x_j^1, x_t^1)$ in space X_1 is a set of edges in space X denoted by e_{jt}^1.

Generally, for space X_i, define a set of its edges as $E_i : (x_j^i, x_t^i \in E_i) \Leftrightarrow \exists x_j^0 \in x_j^i, x_t^0 \in x_t^i, f(x_j^0, x_t^0) \geq w_{i+1}$. Edge $e(x_j^i, x_t^i)$ is represented by $e(x_j^i, x_t^i) = ((x_j^0, p_1(x_j^0), \ldots, p_i(x_j^0) = x_j^i), (x_t^0, p_1(x_t^0), \ldots, p_i(x_t^0) = x_t^i)) | \forall x_j^0 \in x_j^i, x_t^0 \in x_t^i, f(x_j^0, x_t^0) \geq w_{i+1}\}$ or denoted by e_{jt}^i.

Finally, we construct $(X_i, E_i),\ i = 0, 1, \ldots, k$, of the quotient spaces.

Definition 4.2 $\forall x, y \in X, w_i$, x and y are called w_i-connected $\Leftrightarrow$ there exists an edge from x to y on space X and its weight is greater than or equal to w_i.

Theorem 4.1 $\forall x = (x_0, x_1, x_2, \ldots, x_k), y = (y_0, y_1, y_2, \ldots, y_k) \in X, w_i$, *$x$ and y are w_i-connected $\Leftrightarrow x_i = y_i$, where $x = (x_0, x_1, \ldots, x_k)$ and $y = (y_0, y_1, \ldots, y_k)$ are the hierarchical codes of x and y, respectively. x_i, and y_i are denoted by the corresponded codes of x and y in the sequence of hierarchical quotient spaces $(X_0, X_1, \ldots, X_k)$.*

Proof Assume $x_i = y_i$. From definition of x_i, and y_i, it's known that x and y belong to the same connected component on space X_i. Then $p_{i-1}(x)$ and $p_{i-1}(y)$are w_i-connected on space (X_{i-1}, E_{i-1}). On the other hand, $p_{i-2}(z), z \in X_{i-2}$ is w_{i-1}-connected on space X_{i-2}. From the "truth preserving" property in quotient space theory [26, 27], $p_{i-2}(x)$ and $p_{i-2}(y)$ are w_i-connected on space X_{i-2}. By using the "truth preserving" property gradually, we have that x and y are w_i-connected on space X.

Contrarily, if x and y are w_i-connected on space X, obviously, we have $p_i(x) = p_i(y)$ on space X_i, i.e., $x_i = y_i$. □

For each element x_m^i in space (X_i, E_i), construct a matrix P_m^i. Assume that element x_m^i is composed of s elements of space X_{i-1}. Construct an $s \times s$ matrix P_m^i as follows:

$$P_m^i(tj) = \begin{cases} e(x_t^{i-1}, x_j^{i-1}),\ (x_t^{i-1}, x_j^{i-1}) \in E_{i-1}, & \text{if } , m = 1, \ldots, n_i; \\ \emptyset, & \text{otherwise.} \end{cases}$$

Thus, the topological structure of space X_i can be represented by a set $\{P_j^i, j = 1, \ldots, m\}$ of matrices.

In conclusion, the procedure for constructing the hierarchical quotient space model of network (X, E) is shown below:

(i) According to equivalence relation $R(w_1)$, the elements (nodes) of a weighted edge graph (X, E) are classified into several equivalence classes. Based on the classification, we have a quotient space (X_1, E_1), $X_1 = \{x_1^1, \ldots, x_{n_1}^1\}$, and its corresponding matrices $P_1^1, \ldots, P_{n_1}^1$.

(ii) According to equivalence relation $R(w_2)$, the elements (nodes) of the quotient space (X_1, E_1) are further classified into several equivalence

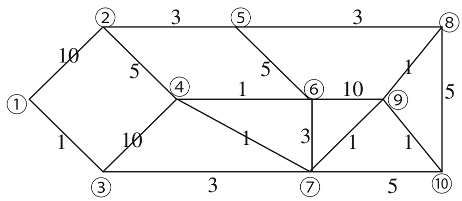

Fig. 1 A weighted network

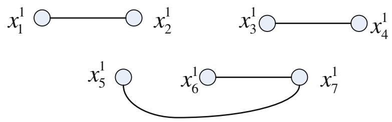

Fig. 2 The quotient space (X_1, E_1)

classes. Then we have a quotient space (X_2, E_2), $X_2 = \{x_1^2, \ldots, x_{n_2}^2\}$, and its corresponding matrices $P_1^2, \ldots, P_{n_2}^2$.

(iii) Generally, according to equivalence relation $R(w_i)$, the elements (nodes) of the space (X_{i-1}, E_{i-1}) are classified into several equivalence classes. We have a quotient space (X_i, E_i), $X_i = \{x_1^i, \ldots, x_{n_i}^i\}$, and its corresponding matrices $P_1^i, \ldots, P_{n_i}^i, 1 \le i \le k$, where k is the number of different weights on edges.

(iv) The construction of quotient spaces will be ended until space (X_j, E_j), $1 \le j \le k$, has only one element or space (X_k, E_k) is obtained.

(v) Ranking the elements of space X_i, we have a sequence of hierarchical quotient spaces $X_0, X_1, \ldots, X_j$. Each element (node) of space X has a hierarchical code $z = (z_0, z_1, \ldots, z_k),\ z \in X$. When $j < k$, $z = (z_0, z_1, \ldots, z_j)$, $z \in X$.

Example 4.1 Find the hierarchical quotient-space model of network in Fig. 1.

In Fig. 1, there are 10 nodes {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}, and a set $w = \{w_1, w_2, w_3, w_4\} = \{10, 5, 3, 1\}$ of weights.

The given space (X_0, E_0) with 10 elements (nodes):

$$X_0 = \{x_1^0 = (1),\ x_2^0 = (2),\ x_3^0 = (3),\ x_4^0 = (4),\ x_5^0 = (5),\ x_6^0 = (6),\ x_7^0 = (7),\ x_8^0 = (8),\ x_9^0 = (9),\ x_{10}^0 = (10)\}$$

From equivalence relation $R(10)$, we have a quotient space X_1 with 7 nodes

$$X_1 = \{x_1^1 = (1,2),\ x_2^1 = (3,4),\ x_3^1 = (5),\ x_4^1 = (6,9),\ x_5^1 = (7),\ x_6^1 = (8),\ x_7^1 = (10)\}.$$

Its corresponding matrices are

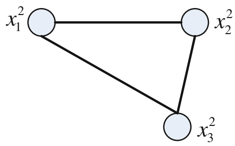

Fig. 3 The quotient space (X_2, E_2)

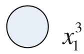

Fig. 4 The quotient space (X_3, E_3)

$$P_1^1 = \begin{pmatrix} 1 & (1,2) \\ & 1 \end{pmatrix},\ P_2^1 = \begin{pmatrix} 1 & (3,4) \\ & 1 \end{pmatrix},\ P_4^1 = \begin{pmatrix} 1 & (6,9) \\ & 1 \end{pmatrix},$$
$$P_3^1 = P_5^1 = P_6^1 = P_7^1 = (1),$$

and the corresponding quotient space (X_1, E_1) as shown in Fig. 2.

From equivalence relation $R(5)$, we have a quotient space X_2 with 3 nodes

$$X_2 = \{x_1^2 = (1,2,3,4),\ x_2^2 = (5,6,9),\ x_3^2 = (7,8,10)\}.$$

Its corresponding matrices

$$P_1^2 = \begin{pmatrix} 1 & ((2,1),(4,2)) \\ & 1 \end{pmatrix}, P_2^2 = \begin{pmatrix} 1 & ((5,3),(6,4)) \\ & 1 \end{pmatrix},$$
$$P_3^2 = \begin{pmatrix} 1 & 0 & ((7,5),(10,7)) \\ & 1 & ((8,6),(10,7)) \\ & & 1 \end{pmatrix},$$

and the corresponding quotient space (X_2, E_2) as shown in Fig. 3.

From equivalence relation $R(3)$, we have a quotient space X_1 with only one node

$x_1^3 = (1,2,3,4,5,6,7,8,9,10)$, its corresponding matrix

$$P_1^3 = \begin{pmatrix} 1 & ((2,1,1),(5,3,2)) & ((3,2,1),(7,5,3)) \\ & 1 & ((5,3,2),(8,6,3))|((6,4,2),(7,5,3)) \\ & & 1 \end{pmatrix},$$

and the corresponding quotient space (X_3, E_3) as shown in Fig. 4.

It's noted that matrix P_j^i is anti-symmetric, i.e., if element $p_{tj}^i = (a,b)$, then $p_{jt}^i = (b,a)$.

Finally, we have the hierarchical codes of each node in space (X_0, E_0) as follows:

$$1 = (1,1,1,1), 2 = (2,1,1,1), 3 = (3,2,1,1), 4 = (4,2,1,1), 5 = (5,3,2,1),$$
$$6 = (6,4,2,1), 7 = (7,5,3,1), 8 = (8,6,3,1), 9 = (9,3,2,1), 10 = (10,7,3,1).$$

5 New Algorithm for Finding Optimal Paths

The optimal path finding procedure begins from the comparison between the last code words in the hierarchical codes of the source node and the destination node to look for the connected path between these two nodes. The procedure carries out from the coarsest quotient space to the finest one gradually until the optimal path is found.

For example, source node $x = (x_0, x_2, \ldots, x_k)$ and destination node $y = (y_0, y_1, y_2, \ldots, y_k)$ in space (X_0, E_0) are given. Compare the last code word x_k with y_k. If $x_k = y_k$, then compare x_{k-1} with y_{k-1} until $x_{i-1} \neq y_{i-1} (0 \leq i \leq k)$ and $x_i = y_i$ so x and y are connected in quotient space (X_{i-1}, E_{i-1}) and equivalent in space (X_i, E_i). Thus, in order to find the connected path between x and y, it's needed to find the connected path between x_{i-1} and y_{i-1} in space (X_{i-1}, E_{i-1}) first. From $P^i_{x_i}$, we may find a connected path $e(x_{i-1}, y_{i-1})$ from x_{i-1} to y_{i-1} in space (X_{i-1}, E_{i-1}). For simplicity, assume that $e(x_{i-1}, y_{i-1}) = (x^1, x^2)$, $x^1 = (x^1_0, \ldots, x^1_{i-1} = x_{i-1})$, and $x^2 = (x^2_0, \ldots, x^2_{i-1} = y_{i-1})$. Inserting x^1 and x^2 into (x, y), we have (x, x^1, x^2, y). Where the $(i-1)$th coordinates of x and x^1 (or x^2 and y) are the same. For x and x^1, the same operation is implemented, i.e., comparing x_{i-2} with x^1_{i-2} until $x_{j-1} \neq x^1_{j-1}$ and $x_j = x^1_j (0 \neq j < i \leq k)$. Finding the connected path in space (X_{j-1}, E_{j-1}), from $P^j_{x_j}$, it's known that $e(x_{j-1}, x^1_{j-1})$ is the connected path from x to x^1. Insert $e(x_{j-1}, x^1_{j-1})$ into x and x^1. The process carries out until the connected path is found on space (X_0, E_0). For x^2 and y, compare x^2_{i-2} with y_{i-2} until $x^2_{j'-1} \neq y_{j'-1}$ and $x^2_{j'} = y_{j'} (0 \leq j' < i \leq k)$. Finding the connected path in space $(X_{j'-1}, E_{j'-1})$, from $P^{j'}_{x_{j'}}$, we know that $e(x^2_{j'-1}, y_{j'-1})$ is the connected path from x^2 to y. Insert $e(x^2_{j'-1}, y_{j'-1})$ into x^2 and y. The procedure continues until the path is found in space (X_0, E_0).

5.1 *The Optimal Path Finding Algorithm*

Given $x = (x_1, \ldots, x_k)$ and $y = (y_1, \ldots, y_k)$ in space (X_0, E_0). Assume that $x_i = y_i, x_j \neq y_j,\ j < i$. Find the w_i-connected edge between x and y on space X_0.

Node x is represented by $x = (p_0(x), p_1(x), \ldots, p_k(x))$, where $p_i : X \to X_i$, $i = 0, 1, \ldots, k$, and $X = X_0$. For $x = (x_1, \ldots, x_k)$ and $y = (y_1, \ldots, y_k)$, by assuming that $x_k = y_k$, x and y are connected in space (X_{k-1}, E_{k-1}) and equivalent in space (X_k, E_k).

(i) From $P^k_{x_k}$, in space (X_{k-1}, E_{k-1}) we have a path $e(x_{k-1}, y_{k-1})$ composed of a_k nodes from x_{k-1} to y_{k-1}. Inserting the a_k nodes into (x, y) in turn, we have a sequence composed of $a_k + 2$ nodes. In the sequence, there is a w_k-edge connected the 2ith with the (2i + 1)th nodes but no edge between the (2i − 1)th and the 2ith nodes, i = 1,…, a_k + 1. Since the (k − 1)th coordinates of x and y are the same, the two nodes are connected in space (X_{k-2}, E_{k-2}).

(ii) Let $k \leftarrow k - 1$, go to step (i).

(iii) The procedure continues until the 0th coordinates of the (2i − 1)th and the 2ith nodes are the same. The sequence of the 0th coordinates is the optimal path.

Example 5.1 Find the optimal path between node 5 and node 7 in Fig. 1.

From Example 4.1, we have a set of quotient spaces as follows: $X_0 = \{x^0_1 = (1),\ x^0_2 = (2),\ x^0_3 = (3),\ x^0_4 = (4),\ x^0_5 = (5),\ x^0_6 = (6),\ x^0_7 = (7), x^0_8 = (8),\ x^0_9 = (9),\ x^0_{10} = (10)\} X_1 = \{x^1_1 = (1, 2),\ x^1_2 = (3, 4),\ x^1_3 = (5),\ x^1_4 = (6, 9),\ x^1_5 = (7),\ x^1_6 = (8) x^1_7 = (10)\} X_2 = \{x^2_1 = (1, 2, 3, 4,),\ x^2_2 = (5, 6, 9),\ x^2_3 = (7, 8, 10)\}.X_3 = \{x^3_1 = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)\}$.

The hierarchical codes of the source node (node 5) and the destination node (node 7) are x = (5, 3, 2, 1) and y = (7, 5, 3, 1), respectively, i.e., (x, y) = ((x_0, x_1, x_2, x_3), (y_0, y_1, y_2, y_3)) = ((5, 3, 2, 1), (7, 5, 3, 1)). By comparing the code words of the hierarchical code of x with that of y, it's known that $x_3 = y_3 = 1$ but $x_2 \neq y_2$. This means that node 5 and node 7 are connected in space (X_2, E_2) and equivalent in space (X_3, E_3), where node 5 and 7 belong to nodes x^2_2 and x^2_3 in space (X_2, E_2), respectively. From matrix P^3_1, we have an w_3-edge between nodes (5, 3, 2) and (8, 6, 3), and an w_3-edge between nodes (6, 4, 2) and (7, 5, 3) in space (X_2, E_2). Inserting these nodes into (x, y), then we have two paths ((5, 3, 2, 1), (5, 3, 2), (8, 6, 3), (7, 5, 3, 1)) and ((5, 3, 2, 1), (6, 4, 2), (7, 5, 3), (7, 5, 3, 1)) from node x to y.

In path ((5, 3, 2, 1), (5, 3, 2), (8, 6, 3), (7, 5, 3, 1)), comparing node (5, 3, 2, 1) with node (5, 3, 2), we have $x_2 = y_2 = 2$, $x_1 = y_1 = 3$, $x_0 = y_0 = 5$. There is a path from node x^3_1 in space (X_3, E_3) to node x^2_2 in space (X_2, E_2). Comparing node (8, 6, 3) with node (7, 5, 3, 1), since $x_2 = y_2 = 3$, $x_1 \neq y_1$, nodes 8 and 7 belong to nodes x^1_6 and x^1_5 in space (X_1, E_1), respectively. The two nodes are connected in space (X_1, E_1) and equivalent in space (X_2, E_2). From matrix P^2_3, it's known that there is an w_2-edge between nodes (8, 6) and (10, 7) and an w_2-edge between nodes (10, 7) and (7, 5) in space (X_1, E_1). Inserting these nodes into the sequence above, we have a path ((5, 3, 2, 1), (5, 3, 2), (8, 6, 3), (8, 6), (10, 7), (10, 7), (7, 5), (7, 5, 3, 1)). Comparing node (8, 6, 3) with node (8, 6), since $x_1 = y_1 = 6$, $x_0 = y_0 = 8$, there is a path from node x_2^3 in space (X_2, E_2) to node x_6^1 in space (X_1, E_1). Nodes (10, 7) and (10, 7) belong to the same node x_7^1 in space (X_1, E_1). Comparing node (7, 5) with node (7, 5, 3, 1), since $x_1 = y_1 = 5$, $x_0 = y_0 = 7$ there is a path from node x_5^1 in space (X_1, E_1) to node x_1^3 in space (X_3, E_3). Finally, we have an optimal path (5, 8, 10, 7) from node 5 to node 7.

Similarly, from sequence ((5, 3, 2, 1), (6, 4, 2), (7, 5, 3), (7, 5, 3, 1)), we have another optimal path (5, 6, 7) from node 5 to node 7.

The path finding procedure is shown in Fig. 5.

6 Experimental Results

In order to compare our algorithm (Hierarchical Quotient Space Model based Algorithm, HQSM algorithm) with other well-known algorithms, we carried out a set of computer simulations. The experimental environment is a java platform. The undirected and weighted networks are generated by random network, small-world network and scale-free network models, respectively. The edge weights are assigned from [31, 32]. The Dijkstra, Floyd and HQSM algorithms are implemented for finding the optimal path of any pair of nodes in the networks.

6.1 The Shortest Path Quality Comparison

For comparison, we replace the edge weight of networks by its reciprocal. Then, the shortest path finding problem is transformed into that of the optimal path finding with the minimal reciprocal sum. We choose random, small-world, and scale-free networks with 100, 200, 300, 400, and 500 nodes as test beds. The Dijkstra, Floyd and HQSM algorithms are implemented for finding the optimal path of any pair of nodes in the networks. The shortest paths found by Dijkstra and Floyd algorithms are always global minimal. The percentage of the shortest paths found by HQSM algorithm that belong to the global minimum in relation to the total number of the shortest paths found is shown in Tables 1, 2 and 3. The number of hierarchical levels used in the quotient-space approach is also shown in the same tables.

6.2 The Computational Complexity Comparison

The undirected and weighted networks with 100, 200, 300, 400 and 500 nodes involve the three network models, i.e., the random, small-world and scale-free

Table 1 Random networks

Number of nodes	Percentage	Number of levels
100	94.60	6
200	95.50	6
300	97.90	6
400	92.60	6
500	97.90	6

Table 2 Small-world networks

Number of nodes	Percentage	Number of levels
100	90.90	5
200	93.60	5
300	85.90	5
400	93.10	5
500	98.70	4

Table 3 Scale-free networks

Number of nodes	Percentage	Number of levels
100	97.10	9
200	98.20	8
300	98.60	7
400	97.40	7
500	99.26	6

Table 4 Total CPU time (in seconds) in the random network

Number of nodes	100	200	300	400	500
HQSM	0.397	1.356	3.112	6.634	12.171
Dijkstra	1.719	9.797	91.141	656.391	1,002.125
Floyd	0.940	4.220	118.630	212.030	511.560

Table 5 Total CPU time (in seconds) in small-world network

Number of nodes	100	200	300	400	500
HQSM	0.640	2.403	6.659	14.33	22.262
Dijkstra	2.758	10.546	87.239	700.540	1,100.200
Floyd	0.790	4.783	132.743	230.412	498.317

Table 6 Total CPU time (in seconds) in scale-free network

Number of nodes	100	200	300	400	500
HQSM	0.437	1.734	4.297	8.187	15.562
Dijkstra	2.045	8.236	86.431	668.400	998.354
Floyd	0.780	4.060	108.598	206.580	466.250

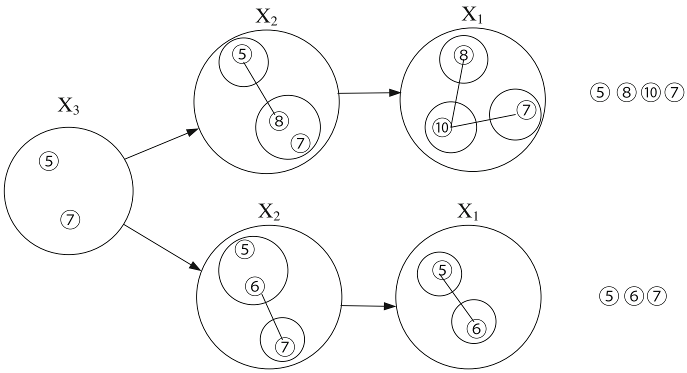

Fig. 5 The optimal paths from node 5 to node 7

networks. The total CPU time for finding the optimal paths of each algorithm is shown in Tables 4, 5 and 6.

The experimental results show that HQSM algorithm outperforms the Dijkstra and Floyd algorithms greatly in saving the computational cost, especially when the networks become larger. On the other hand, Dijkstra and Floyd algorithms need large storage, it's quite difficult to implement the algorithms in networks with more than 500 nodes. In HQSM, the algorithm looks for the optimal paths from the coarse level to the fine one via the hierarchical quotient space model. So it always chooses the high weight edges with high priority. For example, in Fig. 5, space (X_1, E_1) contains $w(5)$-edges ($5 \to 6$, $8 \to 10 \to 7$) and space (X_2, E_2) contains $w(3)$-edges ($5 \to 8$, $6 \to 7$). HQSM looks for the optimal paths from space (X_2, E_2) to space (X_1, E_1). Then the paths ($5 \to 6 \to 7$, and $5 \to 8 \to 10 \to 7$) are found. But Dijkstra algorithm visits nodes 2, 4, 6, 8 first, then nodes 9, 10, 7, and finally finds the optimal path $5 \to 6 \to 7$. Floyd algorithm finds the optimal paths between any pair of nodes by searching all nodes and edges throughout the networks. Although HQSM algorithm can only find the quasi-optimal paths generally, the experimental results show that more than ninety percent of the shortest paths found by HQSM algorithm is global minimal.

7 Conclusion

In this paper, we present a quotient space representation method for problem solving. Based on the method, a problem is represented by a triplet (X, f, T). It enables us to describe different structures, attribute functions and operations on a domain. Especially, it offers a tool for depicting different grain-size worlds.

When (X, T) is a topologic or a semi-order space, we discuss how to construct the quotient topology and quotient (pseudo) semi-order on its corresponding quotient space [X] and after the construction what kind of quotient structures that we can have. We prove three important theorems of different granules, i.e. to preserve false property theorem and to preserve true property theorem.

We supply an example of solving problem by different granule worlds—the shortest path of a complex network. The example indicates that to describe or present a complicated object is equal to construct quotient space. In quotient set [X], the complexity to solve a problem is lower than X. So we have a new solution method to analysis a big data based on the quotient space theory.

References

1. Zhang, B., Zhang, L.: Theory and Applications of Problem Solving. North Holland (1992)
2. Zhang, B., Zhang, L.: Theory and Applications of Problem Solving. Tsinghua University Publisher, Beijing (1990). (in Chinese)
3. Yao, Y.Y.: Perspectives of granular computing. In: Proceedings of 2005 IEEE International Conference on Granular Computing, vol. 1, pp. 85–90 (2005)
4. Bargiela, A., Pedrycz, W.: Granular Computing: An Introduction. Kluwer Academic Publishers, Boston (2002)
5. Dupre, J.: The Discorder of Things, Mataphysical Foundations of the Disunity of Science. Harvard University Press, Cambridge (1993)
6. Foster, C.L.: Algorithms, Abstraction and Implementation: Levels of Detail in Cognitive Science. Academic Press, London (1992)
7. Giunchglia, F., Walsh, T.: A theory of abstraction. Artif. Intell. **56**, 323–390 (1992)
8. Hobbs, J.R.: Granularity. In: Proceedings of the 9th International Joint Conference on Artificial Intelligence, pp. 432–435 (1985)
9. Inuiguchi, M., Hirano, S., Tsumoto, S. (eds.): Rough Set Theory and Granular Computing. Springer, Berlin (2003)
10. Pedrycz, W., Bargiela, A.: An optimization of allocation of information granularity in the interpretation of data structures toward granular fuzzy clustering. IEEE Trans. Syst. Man Cybern. B Cybern. **42**(3), 582–590 (2012)
11. Pedrycz, A., Hirota, K., Pedrycz, W., Dong, F.: Granular representation and granular computing with fuzzy sets. Fuzzy Sets Syst. **203**, 17–32 (2012)
12. Pedrycz, W., Song, M.: Granular fuzzy models: a study in knowledge management in fuzzy modeling. Int. J. Approx. Reasoning **53**(7), 1061–1079 (2012)
13. Pedrycz, W., Bargiela, A.: An optimization of allocation of information granularity in the interpretation of data structures: toward granular fuzzy clustering. IEEE Trans. Syst. Man Cybern. B **42**(3), 582–590 (2012)
14. Pedrycz, W., Homenda, W.: Building the fundamentals of granular computing: a principle of justifiable granularity. Appl. Soft Comput. **13**(10), 4209–4218 (2013)

15. Chen, S.-M., Yang, M.-W., Lee, L.-W., Yang, S.-W.: Fuzzy multiple attributes group decision-making based on ranking interval type-2 fuzzy sets. Expert Syst. Appl. **39**(5), 5295–5308 (2012)
16. Chen, S.M., Randyanto, Y.: A novel similarity measure between intuitionistic fuzzy sets and its applications. Int. J. Pattern Recognit. Artif. Intell. **27**(7), 1350021-1–1350021-34 (2013)
17. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning About Data. Kluwer Academic Publishers, Dordrecht (1991)
18. Pawlak, Z.: Rough sets. Int. J. Comput. Inform. Sci. **11**, 341–356 (1982)
19. Pawlak, Z.: Rough Sets, Theoretical Aspects of Reasoning About Data. Kluwer Academic Publishers, Dordrecht (1991)
20. Pawlak, Z.: Granularity of knowledge, indiscernibility and rough sets. In: Proceedings of 1998 IEEE International Conference on Fuzzy Systems, pp. 106–110 (1998)
21. Zhang, L., Zhang, B.: Quotient Space Theory and Granule computing. CRSSC'2003, Chongqing, China, pp. 1–3 (2003) (in Chinese)
22. Zhang, L., Zhang, B.: Theory of fuzzy quotient space (methods of fuzzy granular computing). J. Softw. **14**(4), 770–776 (2003). (in Chinese)
23. Zhang, Y., Zhang, L., Wu, T.: The representation of different granular worlds: a quotient space. Chin. J. Comput. **27**(3), 328–333 (2004). (in Chinese)
24. Zhang, L., Zhang, B.: The analysis of system performances based on quotient space granular computing. J. Comput. Sci. **31**(10A), 6–9 (2004). (in Chinese)
25. Cheng, W., Shi, Y., Zhang, Y.: Application of quotient space theory in yield prediction. J. Comput. Eng. Appl. **43**(13), 197–199 (2007). (in Chinese)
26. He, F., Zhang, Y., Zhao, S.: The method for the optimal path of complex network based on quotient space hierarchy model (MOCQ). In: Science Paper Online (2007). http://www.paper.edu.cn/releasepaper/content/200712-610 (in Chinese)
27. He, F., Zhang, Y., Chen, J., Zhao, S.: Dynamic information analysis model based on quotient space topology. In: Science Paper Online, vol 5(2), pp. 124–127 (2010) (in Chinese)
28. Zadeh, L.A.: Towards a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. Fuzzy Sets Syst. **19**, 111–127 (1997)
29. Zadeh, L.A.: Fuzzy sets and information granulation. In: Gupta, M., Ragade, R.K., Yager, R.R. (eds.) Advances in Fuzzy Set Theory and Applications. North-Holland Publishing Company, Amsterdam (1979)
30. Zadeh, L.A.: Some reflections on soft computing, granular computing and their roles in the conception, design and utilization of information/intelligent systems. Soft. Comput. **2**, 23–25 (1998)
31. Larson, R., Odoni, A.: Shortest paths between all pairs of nodes. In: Urban Operations Research (1981)
32. Pemmaraju, S., Skiena, S.: All-pairs shortest paths. In: Computational Discrete Mathematics: Combinatorics and Graph Theory in Mathematica, pp 330–331. Cambridge University Press, Cambridge (2003)