Chapter 1
Symmetries: A General Approach to Integrated Uncertainty Management

Vladik Kreinovich, Hung T. Nguyen, and Songsak Sriboonchitta

Abstract We propose to use symmetries as a general approach to maintaining different types of uncertainty, and we show how the symmetry approach can help, especially in economics-related applications.

1.1 Why Symmetries

Formulation of the problem. Our knowledge is rarely complete, we rarely have absolutely certainty. Uncertainty is present in different areas of knowledge. As a result, in many different areas of knowledge, different techniques and approaches have been developed to describe and process uncertainty. For example, in logical-type descriptions of knowledge typical for expert systems and Artificial Intelligence, formalisms like probabilistic logic and fuzzy logic have been developed to process uncertainty. In engineering-oriented probability-type descriptions, probability-related approaches have been developed such as the Dempster-Shafer approach, imprecise probabilities approach, etc.

To solve complex real-life problems, we must takes into account knowledge form different areas. Since these different pieces of knowledge come with uncertainty, we must therefore jointly manage different types of uncertainty.

Vladik Kreinovich
University of Texas at El Paso, El Paso, TX 79968, USA, e-mail: vladik@utep.edu and Faculty of Economics, Chiang Mai University, Chiang Mai 50200, Thailand

Hung T. Nguyen
New Mexico State University, Las Cruces NM 88003, USA, e-mail: hunguyen@nmsu.edu and Faculty of Economics, Chiang Mai University, Chiang Mai 50200, Thailand

Songsak Sriboonchitta
Faculty of Economics, Chiang Mai University, Chiang Mai 50200, Thailand, e-mail: songsak@econ.cmu.ac.th

We therefore need a general approach that would take care of different types of uncertainty.

Symmetry: a fundamental property of the physical world. The reason why we are gaining and processing knowledge is that we want to predict the processes of the physical world, predict the results of different possible actions – and thus, select the action whose results are most beneficial.

On the fundamental level, the very possibility to predict the processes and the results of different actions comes from the fact that we have observed similar situations, we remember the outcomes of these similar situations, and we expect that the outcomes will be similar.

For example, if in the past, we dropped a ball several times and every time, it fell down, then in a new situation we expect the ball to fall down as well. In the past, we may have been at different locations, at different moments of time, oriented differently, but the results were the same. Thus, we conclude that the outcome of this simple drop-the-ball experiment will be the same.

In mathematical terms, the similarity between different situations corresponds to *symmetry*, and the fact that the result is the same for similar situations is usually described as *invariance.*

In these terms, we can say, e.g., that the results of the "drop-the-ball" gravitational experiment are invariant relative to shifting the location, rotating (= changing orientation), and shifting in time.

The notion of symmetry is not only methodologically fundamental: symmetries are one of the main tools of modern physics; see, e.g., [4].

Because of the fundamental nature of symmetries in describing the physical world, it is reasonable to try to use symmetries for describing uncertainty as well.

What is known. The idea of symmetry can indeed explain the basic formulas of different uncertainty formalisms; see, e.g., [8]. For example, natural symmetries explain the most widely used t-norms and t-conorms in fuzzy logic, most widely used non-linear activation functions in neural networks, etc.

What we do in this paper. In this paper, we show that not only the basic formulas, but many other aspects of uncertainty can be explained in terms of symmetries: heuristic and semi-heuristic approaches can be justified by appropriate natural symmetries, and symmetries can help in designing optimal algorithms.

1.2 Symmetries Help in Explaining Existing Algorithms

Practical need for uncertainty propagation. In many practical situations, we are interested in the value of a quantity y which is difficult or even impossible to measure directly. To estimate this difficult-to-measure quantity y, we measure or estimate related easier-to-measure quantities $x_1, \ldots, x_n$ which are related to the desired quantity y by a known relation $y = f(x_1, \ldots, x_n)$. Then, we apply the relation f to the estimates $\widetilde{x}_1, \ldots, \widetilde{x}_n$ for x_i and produce an estimate $\widetilde{y} = f(\widetilde{x}_1, \ldots, \widetilde{x}_n)$ for the desired quantity y.

In the simplest cases, the relation $f(x_1, \ldots, x_n)$ may be an explicit expression: e.g., if we know the current x_1 and the resistance x_2, then we can measure the voltage y by using Ohm's law $y = x_1 \cdot x_2$. In many practical situations, the relation between x_i and y is much more complicated: the corresponding algorithm $f(x_1, \ldots, x_n)$ is not an explicit expression, but a complex algorithm for solving an appropriate non-linear equation (or system of equations).

Estimates are never absolutely accurate:

- measurements are never absolutely precise, and
- expert estimates can only provide approximate values of the directly measured quantities $x_1, \ldots, x_n$.

In both cases, the resulting estimates $\widetilde{x}_i$ are, in general, different from the actual (unknown) values x_i. Due to these estimation errors $\Delta x_i \stackrel{\text{def}}{=} \widetilde{x}_i - x_i$, even if the relation $f(x_1, \ldots, x_n)$ is exact, the estimate $\widetilde{y} = f(\widetilde{x}_1, \ldots, \widetilde{x}_n)$ is different from the actual value $y = f(x_1, \ldots, x_n)$: $\Delta y \stackrel{\text{def}}{=} \widetilde{y} - y \neq 0$.

(In many situations, when the relation $f(x_1, \ldots, x_n)$ is only known approximately, there is an additional source of the approximation error in y caused by the uncertainty in knowing this relation.)

It is therefore desirable to find out how the uncertainty Δx_i in estimating x_i affects the uncertainty Δy in the desired quantity, i.e., how the uncertainties Δx_i *propagate* via the algorithm $f(x_1, \ldots, x_n)$.

Propagation of probabilistic uncertainty. Often, we know the probabilities of different values of Δx_i. For example, in many cases, we know that the approximation errors Δx_i are independent normally distributed with zero mean and known standard deviations σ_i; see, e.g., [10].

In this case, we can use known statistical techniques to estimate the resulting uncertainty Δy in y. For example, since we know the probability distributions, we can simulate them in the computer, i.e., use the Monte-Carlo simulation techniques to get a sample population $\Delta y^{(1)}, \ldots, \Delta y^{(N)}$ of the corresponding errors Δy. Based on this sample, we can then estimate the desired statistical characteristics of the desired approximation error Δy.

Propagation of interval uncertainty. In many other practical situations, we do not know these probabilities, we only know the upper bounds Δ_i on the (absolute values of) the corresponding measurement errors Δx_i: $|\Delta x_i| \leq \Delta$.

In this case, based on the known approximation $\widetilde{x}_i$, we can conclude that the actual (unknown) value of i-th auxiliary quantity x_i can take any value from the interval

$$\mathbf{x}_i = [\widetilde{x}_i - \Delta_i, \widetilde{x}_i + \Delta_i]. \tag{1.1}$$

To find the resulting uncertainty in y, we must therefore find the range $\mathbf{y} = [\underline{y}, \overline{y}]$ of possible values of y when $x_i \in \mathbf{x}_i$:

$$\mathbf{y} = f(\mathbf{x}_1, \ldots, \mathbf{x}_n) \stackrel{\text{def}}{=} \{f(x_1, \ldots, x_n) \,|\, x_1 \in \mathbf{x}_1, \ldots, x_n \in \mathbf{x}_n\}.$$

Computations of this range under interval uncertainty is called *interval computations*; see, e.g., [5].

Comment. It is well known that processing fuzzy uncertainty can be reduced to processing interval uncertainty: namely, the α-cut $\mathbf{y}(\alpha)$ for $y = f(x_1, \ldots, x_n)$ is equal to the range $f(\mathbf{x}_1(\alpha), \ldots, \mathbf{x}_n(\alpha))$; see, e.g., [9].

Linearization. In many practical situations, the approximation errors $\Delta x_i = \widetilde{x}_i - x_i$ are small. In such situations, we can expand the expression for $\Delta y = \widetilde{y} - y$ in Taylor series in Δx_i and keep only the linear terms in this expansion. In this case, we get $\Delta y = c_1 \cdot \Delta x_1 + \ldots + c_n \cdot \Delta x_n$, where $c_i \stackrel{\text{def}}{=} \dfrac{\partial f}{\partial x_i}(\widetilde{x}_1, \ldots, \widetilde{x}_n)$. So if $|\Delta x_i| \le \Delta$, then $|\Delta y| \le \Delta$, where

$$\Delta = |c_1| \cdot \Delta_1 + \ldots + |c_n| \cdot \Delta_n. \tag{1.2}$$

For complex f, we can find c_i by numerical differentiation. To estimate n partial derivatives, we need n calls to f. For large n and complex f, this is too time-consuming.

Cauchy deviate method. For large n, we can further reduce the number of calls to f if we use a special technique of Cauchy-based Monte-Carlo simulations, which enables us to use a fixed number of calls to f (≈ 200) for all possible values n; see, e.g., [6]. This method uses *Cauchy distribution* – i.e., probability distributions with the probability density $\rho(z) = \dfrac{\Delta}{\pi \cdot (z^2 + \Delta^2)}$; the value Δ is called the *(scale) parameter* of this distribution.

Cauchy distribution has the following property that we will use: if $z_1, \ldots, z_n$ are independent random variables, and each of z_i is distributed according to the Cauchy law with parameter Δ_i, then their linear combination

$$z = c_1 \cdot z_1 + \ldots + c_n \cdot z_n \tag{1.3}$$

is also distributed according to a Cauchy law, with a scale parameter $\Delta = |c_1| \cdot \Delta_1 + \ldots + |c_n| \cdot \Delta_n$.

Therefore, if we take random variables δ_i which are Cauchy distributed with parameters Δ_i, then the value

$$\delta \stackrel{\text{def}}{=} f(\widetilde{x}_1, \ldots, \widetilde{x}_n) - f(\widetilde{x}_1 - \delta_1, \ldots, \widetilde{x}_n - \delta_n) =$$

$$c_1 \cdot \delta_1 + \ldots + c_n \cdot \delta_n \tag{1.4}$$

is Cauchy distributed with the desired parameter $\Delta = \sum\limits_{i=1}^{n} |c_i| \cdot \Delta_i$.

Need for intuitive explanation. The Cauchy deviate method is one of the most efficient techniques for processing interval and fuzzy data. However, this method has a serious drawback: while the corresponding technique is mathematically valid, it is somewhat counterintuitive – we want to analyze errors which are located *instead* a given interval $[-\Delta, \Delta]$, but this analysis use Cauchy simulated errors which are located, with a high probability, *outside* this interval.

It is therefore desirable to come up with an intuitive explanation for this technique.

Our main idea: use neurons. Our explanation comes from the idea promoted by Paul Werbos, the author of the backpropagation algorithm for training neural networks. Traditionally, neural networks are used to simulate a deterministic dependence; Paul Werbos suggested that the same neural networks can be used to describe stochastic dependencies as well – if as one of the inputs, we take a standard random number r uniformly distributed on the interval $[0, 1]$; see, e.g., [**?**] and references therein.

In view of this idea, as a natural probability distribution, we can take the result of applying a neural network to this random number. The simplest case is when we have a single neuron. In this case, we apply the activation (input-output) function $f(y)$ corresponding to this neuron to the random number r.

In [8], we described all activation functions $s(x)$ which are optimal with respect to reasonable symmetry-based criteria. It turns out that all such functions have the form $a + b \cdot s_0(K \cdot y + l)$, where $s_0(y)$ is either a linear function, or a fractional-linear function, or $s_0(y) = \exp(y)$, or the logistic (sigmoid) function $s_0(y) = 1/(1 + \exp(-y))$, or $s_0(y) = \tan(y)$. The logistic function is indeed the most popular activation function for actual neural networks, but others are also used. For our purpose, we will use the tangent function: its application of the tangent function to the standard random number r indeed leads to the desired Cauchy distribution.

1.3 Symmetries Help in Designing Optimal Algorithms

Symmetries not only help to find the appropriate representations of uncertainty and appropriate formulas for processing uncertainty: symmetries also help to select the optimal algorithms for implementing the corresponding mathematical formulas.

Fixed points: a practical problem. In many real-life situations, we have dynamical situations which eventually reach an equilibrium.

For example, in *economics*, when a situation changes, prices start changing (often fluctuating) until they reach an equilibrium between supply and demand.

In *transportation*, when a new road is built, some traffic moves to this road to avoid congestion on the other roads; this causes congestion on the new road, which, in its turn, leads drivers to go back to their previous routes, etc. [11].

To describe the problem of finding the *equilibrium* state(s), we must first be able to describe *all possible* states. In this paper, we assume that we already have such a description, i.e., that we know the set X of all possible states.

We must also be able to describe the fact that many states $x \in X$ are not equilibrium states. For example, if the price of some commodity (like oil) is set up too high, it will become profitable to explore difficult-to-extract oil fields; as a new result, the supply of oil will increase, and the prices will drop.

Similarly, as we have mentioned in the main text, if too many cars move to a new road, this road may become even more congested than the old roads initially were, and so the traffic situation will actually decrease – prompting people to abandon this new road.

To describe this instability, we must be able to describe how, due to this instability, the original state x gets transformed in the next moment of time. In other words, we assume that for every state $x \in X$, we know the corresponding state $f(x)$ at the next moment of time.

For non-equilibrium states x, the change is inevitable, so we have $f(x) \neq x$. The equilibrium state x is the state which does not change, i.e., for which $f(x) = x$. Thus, we arrive at the following problem: We are given a set X and a function $f : X \to X$; we need to find an element x for which $f(x) = x$.

In mathematical terms, an element x for which $f(x) = x$ is called a *fixed point* of the mapping f. So, there is a practical need to find fixed points.

The problem of computing fixed points. Since there is a practical need to compute the fixed points, let us give a brief description of the existing algorithms for computing these fixed points; see, e.g., [1].

Straightforward algorithm: Picard iterations. At first glance, the situation seems very simple and straightforward. We know that if we start with a state x at some moment of time, then in the next moment of time, we will get a state $f(x)$. We also know that eventually, we will get an equilibrium. So, a natural thing to do is to simulate how the actual equilibrium will be reached.

In other words, we start with an arbitrary (reasonable) state x_0. After we know the state x_k at the moment k, we predict the state x_{k+1} at the next moment of time as $x_{k+1} = f(x_k)$. This algorithm is called *Picard iterations* after a mathematician who started efficiently using it in the 19 century.

If the equilibrium is eventually achieved, i.e., if in real life the process converges to an equilibrium point x, then Picard's iterations are guaranteed to converge. Their convergence may be somewhat slow – since they simulate all the fluctuations of the actual convergence – but eventually, we get convergence.

Situations when Picard's iterations do not converge: economics. In some important practical situations, Picard iterations do not converge.

The main reason is that in practice, we can have panicky fluctuations which prevent convergence. Of course, one expects fluctuations. For example, if the price of oil is high, then it will become profitable for companies to explore and exploit new oil fields. As a result, the supply of oil will drastically increase, and the price of oil will go down. Since this is all done in a unplanned way, with different companies making very rough predictions, it is highly probable that the resulting oil supply will exceed the demand. As a result, prices will go down, oil production in difficult-to-produce oil areas will become unprofitable, supply will go down, etc.

Such fluctuations have happened in economics in the past, and sometimes, not only they did not lead to an equilibrium, they actually led to deep economic crises.

How can we handle these situation: a natural practical solution. If the natural Picard iterations do not converge, this means that in practice, there is too much of a fluctuation. When at some moment k, the state x_k is not an equilibrium, then at the next moment of time, we have a state $x_{k+1} = f(x_k) \neq x_k$. However, this new state x_{k+1} is an not necessarily closer to the equilibrium: it "over-compensates" by going too far to the other side of the desired equilibrium.

For example, we started with a price x_k which was too high. At the next moment of time, instead of having a price which is closer to the equilibrium, we may get a new price x_{k+1} which is too low – and may even be further away from the equilibrium than the previous price.

In practical situations, such things do happen. In this case, to avoid such weird fluctuations and to guarantee that we eventually converge to the equilibrium point, a natural thing is to "dampen" these fluctuations: we know that a transition from x_k to x_{k+1} has gone too far, so we should only go "halfway" (or even smaller piece of the way) towards x_{k+1}.

How can we describe it in natural terms? In many practical situations, there is a reasonable linear structure on the set X on all the states, i.e., X is a linear space. In this case, going from x_k to $f(x_k)$ means adding, to the original state x_k, a displacement $f(x_k) - x_k$. Going halfway would then mean that we are only adding a half of this displacement, i.e., that we go from x_k to $x_{k+1} = x_k + \frac{1}{2} \cdot (f(x_k) - x_k)$, i.e., to

$$x_{k+1} = \frac{1}{2} \cdot x_k + \frac{1}{2} \cdot f(x_k). \tag{1.5}$$

The corresponding iteration process is called *Krasnoselskii iterations*. In general, we can use a different portions $\alpha \neq 1/2$, and we can also use different portions α_k on different moments of time. In general, we thus go from x_k to $x_{k+1} = x_k + \alpha_k \cdot (f(x_k) - x_k)$, i.e., to

$$x_{k+1} = (1 - \alpha_k) \cdot x_k + \alpha_k \cdot f(x_k). \tag{1.6}$$

These iterations are called *Krasnoselski-Mann iterations.*

Practical problem: the rate of convergence drastically depends on α_i. The above convergence results show that under certain conditions on the parameters α_i, there is a convergence. From the viewpoint of guaranteeing this convergence, we can select any sequence α_i which satisfies these conditions. However, in practice, different choice of α_i often result in drastically different rate of convergence.

To illustrate this difference, let us consider the simplest situation when already Picard iterations $x_{n+1} = f(x_n)$ converge, and converge monotonically. Then, in principle, we can have the same convergence if instead we use Krasnoselski-Mann iterations with $\alpha_n = 0.01$. Crudely speaking, this means that we replace each original step $x_n \to x_{n+1} = f(x_n)$, which bring x_n directly into x_{n+1}, by a hundred new smaller steps. Thus, while we still have convergence, we will need 100 times more iterations than before – and thus, we require a hundred times more computation time.

Since different values α_i lead to different rates of convergence, ranging from reasonably efficient to very inefficient, it is important to make sure that we select *optimal* values of the parameters α_i, values which guarantee the fastest convergence.

Idea: from the discrete iterations to the continuous dynamical system. In this section, we will describe the values α_i which are optimal in some reasonable sense. To describe this sense, let us go back to our description of the dynamical situation. In the above text, we considered observations made at discrete moments of time; this is why we talked about current moment of time, next moment of time, etc. In precise terms, we considered moments t_0, $t_1 = t_0 + \Delta t$, $t_2 = t_0 + 2\Delta t$, etc.

In principle, the selection of Δt is rather arbitrary. For example, in terms of prices, we can consider weekly prices (for which Δt is one week), monthly prices, yearly prices, etc. Similarly, for transportation, we can consider daily, hourly, etc. descriptions. The above discrete-time description is, in effect, a discrete approximation to an actual continuous-time system.

Similarly, Krasnoselski-Mann iterations $x_{k+1} - x_k = \alpha_k \cdot (f(x_k) - x_k)$ can be viewed as a discrete-time approximations to a continuous dynamical system which leads to the desired equilibrium. Specifically, the difference $x_{k+1} - x_k$ is a natural discrete analogue of the derivative $\dfrac{dx}{dt}$, the values α_k can be viewed as discretized values of an unknown function $\alpha(t)$, and so the corresponding continuous system takes the form

$$\frac{dx}{dt} = \alpha(t) \cdot (f(x) - x). \tag{1.7}$$

A discrete-time system is usually a good approximation to the corresponding continuous-time system. Thus, we can assume that, vice versa, the above continuous system is a good approximation for Krasnoselski-Mann iterations.

In view of this fact, in the following text, we will look for an appropriate (optimal) continuous-time system (1.7).

Scale invariance: natural requirement on a continuous-time system. In deriving the continuous system (1.7) from the formula for Krasnoselski-Mann iterations, we assumed that the original time interval Δt between the two consecutive iterations is 1. This means, in effect, that to measure time, we use a scale in which this interval Δt is a unit interval.

As we have mentioned earlier, the choice of the time interval Δt is rather arbitrary. If we make a different choice of this discretization time interval $\Delta t' \neq \Delta t$, then we would get a similar dynamical system, but described in a different time scale, with a different time interval $\Delta t'$ taken as a measuring unit. As a result of "de-discretizing" this new system, we would get a different continuous system of type (1.7) – a system which differs from the original one by a change in scale.

In the original scale, we identified the time interval Δt with 1. Thus, the time t in the original scale means physical time $T = t \cdot \Delta t$. In the new scale, this same physical time corresponds to the time

$$t' = \frac{T}{\Delta t'} = t \cdot \frac{\Delta t}{\Delta t'}. \tag{1.8}$$

If we denote by $\lambda = \dfrac{\Delta t'}{\Delta t}$ the ratio of the corresponding units, then we conclude that the time t in the original scale corresponds to the time $t' = t/\lambda$ in the new scale. Let us describe the system (1.7) in terms of this new time coordinate t'. From the above formula, we conclude that $t = \lambda \cdot t'$; substituting $t = \lambda \cdot t'$ and $dt = \lambda \cdot dt'$ into the formula (1.7), we conclude that

$$\frac{1}{\lambda} \cdot \frac{dx}{dt'} = \alpha(\lambda \cdot t') \cdot (f(x) - x), \tag{1.9}$$

i.e., that

$$\frac{dx}{dt'} = (\lambda \cdot \alpha(\lambda \cdot t')) \cdot (f(x) - x). \tag{1.10}$$

It is reasonable to require that the optimal system of type (1.7) should not depend on what exactly time interval Δt we used for discretization.

Conclusion: optimal Krasnoselski-Mann iterations correspond to $\alpha_k = c/k$. Since a change of the time interval corresponds to re-scaling, this means the system (1.7) must be scale-invariant, i.e., to be more precise, the system (1.10) must have exactly the same form as the system (1.7) but with t' instead of t, i.e., the form

$$\frac{dx}{dt'} = \alpha(t') \cdot (f(x) - x). \tag{1.11}$$

By comparing the systems (1.10) and (1.11), we conclude that we must have $\lambda \cdot \alpha(\lambda \cdot t') = a(t')$ for all t' and λ. In particular, if we take $\lambda = 1/t'$, then we get $\alpha(t') = \dfrac{\alpha(1)}{t'}$, i.e., $\alpha(t') = c/t'$ for some constant c $(= \alpha(1))$.

With respect to the corresponding discretized system, this means that we take $\alpha_k = \alpha(k) = c/k$.

This selection works well. Our experiments on transportation problems confirmed that this procedure converges [2, 3].

The choice $a_k = 1/k$ have been successfully used in other applications as well; see, e.g., [12] and references therein.

Acknowledgments. This work was supported in part by the Faculty of Economics at Chiang Mai University, Thailand, by the National Science Foundation grants HRD-0734825 and DUE-0926721, and by Grant 1 T36 GM078000-01 from the National Institutes of Health.

References

1. V. Berinde, *Iterative approximation of fixed points* (Editura Efemeride, Baia Mare, 2002)
2. R. Cheu, V. Kreinovich, Y.-C. Chiu, R. Pan, G. Xiang, S.V. Bhupathiraju, and S.R. Manduva, *Strategies for Improving Travel Time Reliability*, Texas Department of Transportation, Research Report 0-5453-R2, 2007
3. R. Cheu, V. Kreinovich, F. Modave, G. Xiang, and T. Magoc, "How to estimate, take into account, and improve travel time reliability in transportation networks", In: R.L. Muhanna and R.L. Mullen (eds.) *Proc. Int'l Workshop on Reliable Engineering Computing REC'08*, Savannah, Georgia, February 20–22, 2008, pp. 289–332
4. R. Feynman, R. Leighton, and M. Sands, *The Feynman Lectures on Physics* (Addison Wesley, Boston, Massachusetts, 2005)
5. L. Jaulin, M. Kieffer, O. Didrit, and E. Walter, *Applied Interval Analysis* (Springer-Verlag, London, 2001)
6. V. Kreinovich and S. Ferson, "A New Cauchy-Based Black-Box Technique for Uncertainty in Risk Analysis", *Reliability Engineering and Systems Safety*, 2004, Vol. 85, No. 1–3, pp. 267–279.
7. V. Kreinovich and H.T. Nguyen, "Towards A Neural-Based Understanding of the Cauchy Deviate Method for Processing Interval and Fuzzy Uncertainty", *Proc. IFSA'2009*, Lisbon, Portugal, July 20–24, 2009, pp. 1264–1269.
8. H.T. Nguyen and V. Kreinovich, *Applications of continuous mathematics to computer science* (Kluwer, Dordrecht, 1997)
9. H.T. Nguyen and E.A. Walker, *A First Course in Fuzzy Logic* (CRC Press, Boca Raton, Florida, 2006)
10. S. Rabinovich, *Measurement Errors and Uncertainties: Theory and Practice* (American Institute of Physics, New York, 2005)
11. Y. Sheffi, *Urban Transportation Networks* (Prentice Hall, Englewood Cliffs, NJ, 1985)
12. Y. Su and X. Qin, "Strong convergence theorems for asymptotically nonexpansive mappings and asymptotically nonexpansive semigroups", *Fixed Point Theory and Applications*, 2006, Article ID 96215, pp. 1–11