

Title: Soft Computing Author: Eva Volná, University of Ostrava, Czech Republic	Název: Soft Computing Autor: Eva Volná, University of Ostrava, Czech Republic	Název: Soft Computing Autor: Eva Volná, University of Ostrava, Czech Republic (My translation)
<p>Soft computing became a formal area of study in computer science in the early 1990s. Prof. Lotfi Aliasker Zadeh has defined <i>soft computing</i> as a number of techniques and methods that deal with real-world situations in the same way that people deal with them. In computer science, soft computing (sometimes referred to as Computational Intelligence - CI) is the use of inexact solutions to computationally hard tasks such as the solution of NP-complete problems, for which there is no known algorithm that can compute an exact solution in polynomial time. Soft computing differs from conventional computing in that, unlike hard computing, it is tolerant of imprecision, uncertainty, partial truth, and approximation. In effect the role model for soft computing is the human mind.</p> <p>Soft computing forms the basis of a considerable amount of machine learning techniques. The principal constituents of soft computing are <i>fuzzy logic</i>, <i>evolutionary computation</i>, and <i>neural networks</i>. Generally</p>	<p>М'які обчислення стали офіційною областю інформатики на початку 1990-х років. Проф. Lotf Aliasker Zadeh визначив м'які обчислення як серію методів і методів, які стосуються реальних ситуацій так само, як і люди. У обчислювальних засобах, м'які обчислення (іноді звані Computational Intelligence CI) використовуються для неточних рішень обчислювальних завдань, таких як розв'язання NP-повних завдань, для яких немає відомого алгоритму, який може обчислити точні рішення в поліноміальний час. М'які обчислення відрізняються від звичайних обчислень тим, що, на відміну від них, він терпимо до неточності, невизначеності, часткової істини та наближення. Модель для м'яких обчислень - це насправді людський розум.</p> <p>М'які обчислення є основою великої кількості методів машинного навчання. Основними компонентами м'яких обчислень є нечітка логіка, еволюційні</p>	<p>М'які обчислення стали загальновизнаною галуззю досліджень в інформатиці на початку 1990-х років. Проф. Lotf Aliasker Zadeh визначив м'які обчислення як сукупність методів і методик, які призначені для опрацювання даних, що стосуються ситуацій реального світу подібно до того, як їх опрацьовують люди. В інформатиці, м'які обчислення (іноді звані Computational Intelligence CI) використовуються для приблизних рішень обчислювально складних задач, таких як розв'язання NP-повних задач, для яких немає відомого алгоритму, за допомогою якого можна знайти точні рішення в поліноміальний час. М'які обчислення відрізняються від звичайних обчислень тим, що, на відміну від останніх, вони припускають приблизні, неточні, часткові рішення та наближення. Внаслідок цього, модель для м'яких обчислень - це насправді людський розум.</p> <p>М'які обчислення є основою великої кількості методів машинного навчання.</p>

<p>speaking, soft computing techniques resemble biological processes more closely than traditional techniques, which are largely based on formal logical systems. Soft computing techniques are intended to complement each other.</p>	<p>обчислення та нейронні мережі. Взагалі кажучи, методи м'яких обчислень більше схожі на біологічні процеси, ніж традиційні методи, особливо на основі формальних логічних систем. М'які обчислювальні методи доповнюють один одного.</p>	<p>Основними компонентами м'яких обчислень є нечітка логіка, еволюційні обчислення та нейронні мережі. Взагалі кажучи, методи м'яких обчислень в більшій мірі нагадують біологічні процеси, ніж традиційні методи, особливо на основі формальних логічних систем. Методи м'яких обчислень орієнтовані на те, щоб доповнювати один одного.</p>
<p>Fuzzy logic. Fuzzy logic is based on the theory of fuzzy sets, which is a generalization of the classical set theory. Classical logic only permits conclusions which are either true or false. In fuzzy logic the truth values of variables may be any real number between 0 and 1 inclusive. Therefore, the membership function is a curve that defines how each point in the input space is mapped to a membership value between 0 and 1. The input space is referred to as the universe of discourse. The universe of discourse X is a continuous space, we usually partition X into several fuzzy sets. These fuzzy sets, which usually carry names that conform to adjectives appearing in our daily linguistic usage, such as "large", "medium" or "small" are called linguistic variables.</p>	<p>Нечітка логіка. Нечітка логіка ґрунтується на теорії нечітких множин, що є узагальненням класичної теорії множин. Класична логіка дозволяє лише висновки, які є або істинними, або помилковими. У нечіткій логіці значення істинності змінних можуть бути будь-якими дійсними числами від 0 до 1 включно. Отже, нечітка функція елемента визначається як крива, яка визначає, як кожна точка у вхідному просторі відображається у значення між 0 і 1. Вхідний простір називається вселенною дискурсу. Всесвіт дискурсу X є безперервним простором і зазвичай поділяється на кілька нечітких множин. Ці нечіткі множини називаються мовними змінними, тому що вони мають назви, які відповідають прикметників, що з'являються на нашому природному мові, таких як "великий", "середній" або "малий".</p>	<p>Нечітка логіка. Нечітка логіка ґрунтується на теорії нечітких множин, що є узагальненням класичної теорії множин. У класичній логіці допускаються лише висновки, які є або істинними, або хибними. У нечіткій логіці значення істинності змінних можуть бути будь-якими дійсними числами від 0 до 1 включно. Отже, нечітка функція елемента визначається як крива, яка визначає, як кожна точка у вхідному просторі відображається у значення між 0 і 1. Вхідний простір називається універсумом дискурсу. Універсум дискурсу X є незперервним простором і зазвичай поділяється на кілька нечітких множин. Ці нечіткі множини називаються мовними змінними, тому що вони мають назви, які відповідають прикметникам, що з'являються у нашому щоденному мовному спілкуванні, таких як "великий", "середній" або "малий".</p>

<p>Creating a system with fuzzy logic contains four steps:</p> <ol style="list-style-type: none"> 1. <i>Rule base</i>: It contains the set of IF-THEN rules provided by the experts to govern the decision making system, on the basis of linguistic information. 2. <i>Fuzzification</i>: It is used to convert inputs i.e. crisp numbers into fuzzy sets. 3. <i>Inference engine</i>: It determines the matching degree of the current fuzzy input with respect to each rule and decides which rules from the rule base are to be fired according to the input field. 4. <i>Defuzzification</i>: It is used to convert the fuzzy sets obtained by inference engine into a crisp value. 	<p>Створення системи з нечіткою логікою відбувається в чотири кроки:</p> <ol style="list-style-type: none"> 1. Основа правила: містить набір правил IF-THEN, які містять мовні змінні. Ця основа була розроблена фахівцями в даній області. 2. Fuzzification: Використовується для перетворення входів, тобто точних чисел, до нечітких множин. 3. Перешкоди: визначає відповідний ступінь приналежності поточного входу до нечіткого множини по відношенню до кожного правила в базі правил і визначає, які правила повинні застосовуватися. 4. Defuzzification: Використовується для перетворення отриманого нечіткого набору у реальне значення. 	<p>Створення системи з нечіткою логікою відбувається в чотири етапи:</p> <ol style="list-style-type: none"> 1. Основа правила: містить набір правил IF-THEN, які надаються фахівцями у даній галузі, і призначені для того, щоб керувати прийняттям рішень із використанням лінгвістичної інформації. 2. Fuzzification: Використовується для перетворення вхідних даних, тобто точних чисел, у нечіткі множини. 3. Машина виводу: визначає ступінь відповідності поточних нечітких вхідних даних відносно до кожного правила в базі правил і визначає, які правила повинні застосовуватися в залежності від вхідного поля даних 4. Defuzzification: Використовується для перетворення отриманих за допомогою машини виводу нечітких множин у реальні значення.
<p>Evolutionary computation. In computer science, evolutionary computation is a family of algorithms for global optimization inspired by biological evolution. In technical terms, they are a family of population-based trial and error problem solvers with a metaheuristic or stochastic optimization character. In evolutionary computation, an</p>	<p>Еволюційні розрахунки. Еволюційні методи належать до сімейства алгоритмів глобальної оптимізації, натхнених біологічною еволюцією. З технічної точки зору, це стохастичні алгоритми оптимізації на основі популяції. У еволюційному розрахунку початкова популяція кандидатських рішень</p>	<p>Еволюційні обчислення. В комп'ютерній науці до еволюційних методів обчислень належить сімейство алгоритмів глобальної оптимізації, що виникли під впливом ідеї біологічної еволюції. З технічної точки зору, це методи розв'язання проблем на основі популяції із використанням спроб і помилок з мета</p>

<p>initial set of candidate solutions is generated and iteratively updated. Each new generation is produced by stochastically removing less desired solutions, and introducing small random changes. In biological terminology, a population of solutions is subjected to natural selection (or artificial selection) and mutation. As a result, the population will gradually evolve to increase in fitness, in this case the chosen fitness function of the algorithm.</p> <p>Evolutionary computation techniques can produce highly optimized solutions in a wide range of problem settings, making them popular in computer science.</p>	<p>генерується випадковим чином і ітеративно оновлюється. Кожне нове покоління створюється стохастично, видаляючи менш бажані рішення і приймаючи невеликі випадкові зміни. У біологічній термінології популяція підлягає природному відбору (або штучному відбору) і мутації. Як наслідок, населення поступово розвивається, щоб підвищити пристосованість своїх рішень. Еволюційне обчислення дозволяє знайти оптимальне рішення для широкого кола проблем, що робить його популярним інструментом у комп'ютерній науці.</p>	<p>евристичними або стохастичною оптимізацією. В еволюційних обчисленнях генерується вихідний набір рішень, що пропонуються, який ітеративно оновлюється. Кожне нове покоління створюється шляхом стохастичного видалення менш бажаних рішень і запровадження невеликих випадкових змін. У біологічній термінології популяція підлягає природному відбору (або штучному відбору) і мутації. Як наслідок, популяція буде поступово розвиватися, щоб підвищити пристосованість своїх рішень, у відповідності до алгоритму обраної функції відповідності.</p> <p>Методи еволюційних обчислень дозволяють знайти високо оптимальні рішення для широкого кола проблем, що робить його популярним інструментом у комп'ютерній науці.</p>
<p>Evolutionary computation proceeds in the following steps:</p> <ol style="list-style-type: none"> 1. Test each chromosome to see how good it is at solving the problem at hand and assign a fitness score accordingly. The fitness score is a measure of how good that chromosome is at solving the problem to hand. 2. Select two members from the current 	<p>Еволюційний розрахунок працює за наступними етапами:</p> <ol style="list-style-type: none"> 1. Протестуйте кожну хромосому в популяції, щоб побачити, наскільки добре вона має справу з проблемою, і призначте відповідне значення. 2. Виберіть двох осіб з поточного населення. Вибір пропорційний величині фітнесу кожної хромосоми. 3. Застосувати оператор схрещування в 	<p>Еволюційні обчислення здійснюються в наступні етапи:</p> <ol style="list-style-type: none"> 1. Протестуйте кожну хромосому в популяції, щоб побачити, наскільки вона добре підходить для вирішення проблеми, відповідно до цього їй призначається певне значення функції відповідності. 2. Виберіть двох членів з поточної популяції. Шанс бути обраними пропорційний величині відповідності

<p>population. The chance of being selected is proportional to the chromosomes fitness. Roulette wheel selection is a commonly used method.</p> <ol style="list-style-type: none"> 3. Dependent on the crossover rate crossover the bits from each chosen chromosome at a randomly chosen point. 4. Step through the chosen chromosomes bits and flip dependent on the mutation rate. <p>Repeat step 2, 3, 4 until a new population has been created.</p>	<p>залежності від ймовірності операції схрещування.</p> <ol style="list-style-type: none"> 4. Застосувати оператор мутації в залежності від ймовірності операції мутації <p>Повторіть кроки 2, 3, 4, доки не буде створено нове населення.</p>	<p>кожної хромосоми. Метод рулетки як правило застосовується для вибору.</p> <ol style="list-style-type: none"> 3. Застосувати оператор схрещування в залежності від ймовірності операції схрещування. 4. Застосувати оператор мутації в залежності від ймовірності операції мутації <p>Повторіть кроки 2, 3, 4, доки не буде створено нову популяцію.</p>
<p>Artificial neural networks. Artificial neural networks are computing systems inspired by the biological neural networks. The neural network itself is not an algorithm, but rather a framework for many different machine learning algorithms to work together and process complex data inputs. Such systems "learn" to perform tasks by considering examples, generally without being programmed with any task-specific rules.</p> <p>An artificial neural network is based on a collection of connected units or nodes called artificial neurons, which loosely model the neurons in a biological brain. Each connection, like the synapses in a biological brain, can transmit a signal from one</p>	<p>Штучні нейронні мережі. Штучні нейронні мережі є обчислювальними системами, натхненні біологічними нейронними мережами. Сама нейронна мережа не є алгоритмом, а основою для багатьох різних алгоритмів машинного навчання, які працюють разом і обробляють складні вхідні дані. Ці системи "навчаються" виконувати завдання на основі прикладів, не запрограмованих відповідно до конкретних правил.</p> <p>Штучна нейронна мережа містить вузли, які називаються штучними нейронами, які вільно моделюють нейрони в біологічному мозку. Будь-який зв'язок між</p>	<p>Штучні нейронні мережі. Штучні нейронні мережі є обчислювальними системами, що виникли по аналогії з біологічними нейронними мережами. Сама нейронна мережа не є алгоритмом, а основою для багатьох різних алгоритмів машинного навчання, які працюють разом і обробляють складні вхідні дані. Ці системи "навчаються" виконувати завдання на основі прикладів, не запрограмованих відповідно до конкретних правил.</p> <p>Штучна нейронна мережа містить вузли, які називаються штучними нейронами, які в деякому наближенні моделюють нейрони біологічного мозку. Будь-який</p>

<p>artificial neuron to another. An artificial neuron that receives a signal can process it and then signal additional artificial neurons connected to it.</p>	<p>нейронами, а також синапсами в біологічному мозку, може передавати сигнал від одного нейрона до іншого. Нейрон, який отримує сигнал, може обробляти його, а потім відправляти його до інших нейронів, з якими вони пов'язані.</p>	<p>зв'язок між нейронами, аналогічно нейронам і синапсам в біологічному мозку може передавати сигнал від одного нейрона до іншого. Нейрон, який отримує сигнал, може обробляти його, а потім відправляти його до інших нейронів, з якими він пов'язаний.</p>
<p>In common implementations, the signal at a connection between artificial neurons are a real number, and the output of each artificial neuron is computed by some non-linear function of the sum of its inputs. Artificial neurons and edges typically have a weight that adjusts as learning proceeds. Typically, artificial neurons are aggregated into layers. Signals travel from the first layer (the input layer), to the last layer (the output layer).</p> <p>The original goal of the ANN approach was to solve problems in the same way that a human brain would. However, over time, attention moved to performing specific tasks, leading to deviations from biology. Artificial neural networks have been used on a variety of tasks, including computer vision, speech recognition, machine translation, noise filtering, playing board and video games and medical diagnosis etc.</p>	<p>У загальних реалізаціях сигнал на зв'язках між нейронами є дійсним числом, і вихід з кожного штучного нейрона обчислюється деякою нелінійною функцією з суми її зважених входів. Нейрональні зв'язки мають визначене значення ваги, яке пристосовується до навчання. Нейрони, як правило, розташовані в шарах. Сигнали переходять від першого шару (вхідний шар) до останнього шару (вихідний шар). Первісна мета штучних нейронних мереж полягала в тому, щоб вирішувати проблеми таким же чином, як і їхній людський мозок. Проте з плином часу увага перейшла до вирішення конкретних проблем, що призвело до відходу від біології. Штучні нейронні мережі використовуються для вирішення комп'ютерного зору, розпізнавання мовлення, машинного перекладу, вилучення шуму, ігор та відеоігор, медичної діагностики тощо.</p>	<p>У звичайній реалізації сигнал, що виникає на зв'язку між нейронами, є дійсним числом, і вихід з кожного штучного нейрона обчислюється деякою нелінійною функцією з суми її зважених входних значень. Штучні нейрони і зв'язки мають зазвичай значення ваги, яке використовується для налаштування по мірі прогресу навчання. Нейрони, як правило, розташовані в шарах. Сигнали переходять від першого шару (вхідний шар) до останнього шару (вихідний шар). Початкова мета створення штучних нейронних мереж полягала в тому, щоб вони вирішували проблеми подібно до того, як і людський мозок. Проте з плином часу увага перейшла до вирішення певних специфічних задач, що призвело до відходу від біології. Штучні нейронні мережі використовуються для вирішення спектру задач, зокрема. комп'ютерного зору, розпізнавання мовлення, машинного перекладу, фільтрації шуму, ігор та</p>

		відеоігор, медичної діагностики тощо.
--	--	---------------------------------------

Title: Soft Computing

Author: Eva Volná, University of Ostrava, Czech Republic

Soft computing became a formal area of study in computer science in the early 1990s. Prof. Lotfi Aliasker Zadeh has defined *soft computing* as a number of techniques and methods that deal with real-world situations in the same way that people deal with them. In computer science, soft computing (sometimes referred to as Computational Intelligence - CI) is the use of inexact solutions to computationally hard tasks such as the solution of NP-complete problems, for which there is no known algorithm that can compute an exact solution in polynomial time. Soft computing differs from conventional computing in that, unlike hard computing, it is tolerant of imprecision, uncertainty, partial truth, and approximation. In effect the role model for soft computing is the human mind.

Soft computing forms the basis of a considerable amount of machine learning techniques. The principal constituents of soft computing are *fuzzy logic*, *evolutionary computation*, and *neural networks*. Generally speaking, soft computing techniques resemble biological processes more closely than traditional techniques, which are largely based on formal logical systems. Soft computing techniques are intended to complement each other.

Fuzzy logic. Fuzzy logic is based on the theory of fuzzy sets, which is a generalization of the classical set theory. Classical logic only permits conclusions which are either true or false. In fuzzy logic the truth values of variables may be any real number between 0 and 1 inclusive. Therefore, the membership function is a curve that defines how each point in the input space is mapped to a membership value between 0 and 1. The input space is referred to as the universe of discourse. The universe of discourse X is a continuous space, we usually partition X into several fuzzy sets. These fuzzy sets, which usually carry names that conform to adjectives appearing in our daily linguistic usage, such as "large", "medium" or "small" are called linguistic variables.

Creating a system with fuzzy logic contains four steps:

1. Rule base: It contains the set of IF-THEN rules provided by the experts to govern the decision making system, on the basis of linguistic information.
2. Fuzzification: It is used to convert inputs i.e. crisp numbers into fuzzy sets.
3. Inference engine: It determines the matching degree of the current fuzzy input with respect to each rule and decides which rules from the rule base are to be fired according to the input field.
4. Defuzzification: It is used to convert the fuzzy sets obtained by inference engine into a crisp value.

Evolutionary computation. In computer science, evolutionary computation is a family of algorithms for global optimization inspired by biological evolution. In technical terms, they are a family of population-based trial and error problem solvers with a metaheuristic or stochastic optimization character. In evolutionary computation, an initial set of candidate solutions is generated and iteratively updated. Each new generation is produced by stochastically removing less desired solutions, and introducing small random changes. In biological terminology, a population of solutions is subjected to natural selection (or artificial selection) and mutation. As a result, the population will gradually evolve to increase in fitness, in this case the chosen fitness function of the algorithm.

Evolutionary computation techniques can produce highly optimized solutions in a wide range of problem settings, making them popular in computer science.

Evolutionary computation proceeds in the following steps:

1. Test each chromosome to see how good it is at solving the problem at hand and assign a fitness score accordingly. The fitness score is a measure of how good that chromosome is at solving the problem to hand.
2. Select two members from the current population. The chance of being selected is proportional to the chromosomes fitness. Roulette wheel selection is a commonly used method.
3. Dependent on the crossover rate crossover the bits from each chosen chromosome at a randomly chosen point.
4. Step through the chosen chromosomes bits and flip dependent on the mutation rate.

Repeat step 2, 3, 4 until a new population has been created.

Artificial neural networks. Artificial neural networks are computing systems inspired by the biological neural networks. The neural network itself is not an algorithm, but rather a framework for many different machine learning algorithms to work together and process complex data inputs. Such systems "learn" to perform tasks by considering examples, generally without being programmed with any task-specific rules.

An artificial neural network is based on a collection of connected units or nodes called artificial neurons, which loosely model the neurons in a biological brain. Each connection, like the synapses in a biological brain, can transmit a signal from one artificial neuron to another. An artificial neuron that receives a signal can process it and then signal additional artificial neurons connected to it.

In common implementations, the signal at a connection between artificial neurons are a real number, and the output of each artificial neuron is computed by some non-linear function of the sum of its inputs. Artificial neurons and edges typically have a weight that adjusts as learning proceeds. Typically, artificial neurons are aggregated into layers. Signals travel from the first layer (the input layer), to the last layer (the output layer).

The original goal of the ANN approach was to solve problems in the same way that a human brain would. However, over time, attention moved to performing specific tasks, leading to deviations from biology. Artificial neural networks have been used on a variety of tasks, including computer vision, speech recognition, machine translation, noise filtering, playing board and video games and medical diagnosis etc.