

## ЗАСТОСУВАННЯ НЕЙРОННИХ МЕРЕЖ ХОПФІЛДА ДЛЯ РОЗВ'ЯЗАННЯ CSP ЗАДАЧ

*У статті здійснено огляд методів, що базуються на нейронній мережі Хопфілда, для розв'язання CSP та FCSP задач. Першу спробу застосувати цей тип нейронних мереж до розв'язання CSP задачі зробив сам Хопфілд, після чого відбулася низка модифікацій запропонованого ним алгоритму. Тобто всі представлені у статті методи є модифікаціями один одного та розвивалися послідовно. Також наведено деякі характеристики методів на основі мережі Хопфілда в порівнянні з іншими (не пов'язаними з нейронними мережами) алгоритмами розв'язання CSP.*

**Ключові слова:** CSP, задача задоволення обмежень, нейронна мережа, нейронна мережа Хопфілда.

У галузі штучного інтелекту існує клас комбінаторних задач, які називаються CSP задачі (Constraint Satisfaction Problems). Вони є потужним засобом для розв'язання практичних задач, які можуть бути спроектовані на множину змінних, пов'язаних між собою обмеженнями. При цьому класична CSP модель є досить примітивною для того, щоб бути прототипом більшості реальних життєвих задач. Інколи буває важко сформулювати обмеження в реальних задачах у жорсткій формі, тобто таким чином, щоб можна було однозначно визначити, задоволене це обмеження чи ні. Інший варіант, при якому можуть виникнути подібні проблеми, – це якщо неповне задоволення обмежень є припустимим. Більше того, в деяких ситуаціях повне задоволення всіх обмежень задачі є неможливим.

Усі ці фактори викликали появу різновиду CSP задач, що став називатися FCSP (Fuzzy Constraint Satisfaction Problem). На відміну від CSP задач, FCSP задачі оперують нечіткою логікою та гнучкими обмеженнями, що передбачає можливість неповного задоволення обмежень у задачі.

На сьогодні існує великий арсенал методів розв'язання CSP задач [1], тоді як кількість алгоритмів для розв'язання FCSP доволі невелика. В основному FCSP розв'язують методом зведення її до CSP задачі чи ряду CSP задач із подальшим їх розв'язанням. Утім, існують й інші методи, зокрема, алгоритми розв'язання FCSP задач на основі нейронних мереж. Традиційно нейронні мережі застосовуються для розв'язання задач прогнозування, класифікації та розпізнавання образів, але історія показує, що була також численна кількість спроб застосувати їх до розв'язання комбінаторних оптимізаційних задач. Багато з цих спроб були вдалими,

і ефективність побудованих методів була порівнюваною з альтернативними підходами [10]. Утім, незважаючи на це, прийнято вважати, що нейронні мережі «не дуже успішні при застосуванні їх для розв'язання оптимізаційних задач і вони не йдуть ні в яке порівняння з найкращими метаевристичними», як-от метод відпалювання, табу-пошук, генетичні алгоритми [15]. Ця стаття покликана реабілітувати авторитет нейронних мереж, принаймні в галузі розв'язання CSP задач.

Історично склалося, що до розв'язання CSP задач використовувалися тільки нейронні мережі з навчанням без учителя. Вони найкраще з усіх підходять для цієї мети, оскільки не потребують генерування навчальних прикладів. Це легко пояснити на простому прикладі: якщо вважати, що на вхід нейронній мережі подається розв'язок CSP задачі, то «хорошим» навчальним прикладом для мережі повинен був би бути допустимий розв'язок CSP, отже, ми би вже мали розв'язок, що очевидно є абсурдом. Утім, це не означає, що нейронні мережі з навчанням з учителем не застосовні до CSP задач, просто CSP задачу треба правильно спроектувати на нейронну мережу.

Найпопулярнішою нейронною мережею для розв'язання CSP задач виявилася мережа Хопфілда, про яку й піде мова у статті.

### CSP задача

Constraint Satisfaction Problem називають задачу, що складається з  $N$  змінних, множини доменів  $A_i$  ( $i = 1 \dots n$ ) цих змінних та множини обмежень, накладених на ці змінні  $C(P_1) \dots C(P_r)$ , де  $C(P_j)$  є множиною значень змінних, пов'язаних цим обмеженням [3].

Наприклад, такою задачею є задача про розфарбування графа, де змінними  $X_1, \dots, X_n$  є вершини графа, а значенням змінної  $X_i$  є колір, у який розфарбовано цю вершину. Обмеження мають вигляд  $C(X_p, X_j) = \{(a, b) : a, b - \text{різні кольори}\}$  для кожної пари  $(X_p, X_j)$  сусідніх вершин графа.

CSP модель побудована на жорсткій логіці, що означає, що обмеження можуть набувати тільки значень 0 (обмеження повністю не задоволене) або 1 (обмеження повністю задоволене). Тобто для отримання допустимого розв'язку такої задачі необхідно повністю задовольнити всі її обмеження.

### FCSP задача

Fuzzy Constraint Satisfaction Problem є різновидом CSP задач. Вона з'явилася в момент, коли потенціал CSP задач було повністю розкрито, і стало очевидно, що CSP являє собою погану модель для проектування на неї реальних задач, де часто неможливо обійтися жорсткими умовами та повним задоволенням усіх обмежень. У такій ситуації на допомогу прийшла нечітка логіка Задеха [21], яка, на відміну від звичайної жорсткої логіки, не обмежує себе двома константами для визначення хибності чи істинності твердження, тобто 0 (false) та 1 (true), а пропонує використовувати для цієї мети цілий інтервал значень  $[0, 1]$ . Також вона пропонує таку математичну структуру, як нечіткі множини, які не складаються із жорстко визначеного набору елементів, а натомість з елементів, які належать їм (множинам) певною мірою.

Застосування концепції нечіткої логіки до CSP задач дозволило розв'язувати, наприклад, несумісні задачі, де всі обмеження не можуть бути задоволені повністю одночасно. Інший приклад – задачі з великою кількістю обмежень (overconstrained), для яких теж зазвичай важко або неможливо знайти розв'язок, застосовуючи жорстку логіку.

Перша спроба формалізувати FCSP, мабуть, належить Дюбуа [5] та Рутткай [16]. Усі наступні спроби були досить схожими на перші, тому розглянемо модель FCSP задач, описану в одній із таких статей [7].

Нехай є простір усіх можливих розв'язків задачі  $U$ , який складається з векторів вигляду  $(x_1, \dots, x_k)$ , де  $k$  – кількість змінних у задачі, а  $x$  набувають усіх можливих значень зі своїх доменів. Введемо функцію «ступеня належності» («membership degree function»)  $A: U \rightarrow [0, 1]$ , яка визначатиме міру належності кожного вектора  $(x_1, \dots, x_k)$  до множини розв'язків, або, точніше, міру, з якою цей розв'язок

задовольняє всі обмеження задачі. У звичайних CSP задачах із жорсткими обмеженнями ця функція має вигляд  $A: U \rightarrow \{0, 1\}$ .

Тоді розв'язком FCSP задачі вважатиметься будь-який можливий розв'язок, що належить до «альфа-зрізу»:

$$A|_{\alpha} = \{u \in U \mid A(u) \geq \alpha\}.$$

Причому спосіб встановлення порога  $\alpha$  або є частиною алгоритму розв'язання задачі, або ж мануально обирається під час розв'язання.

### Нейронні мережі Хопфілда

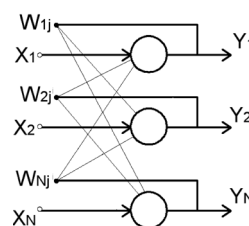


Рис. 1. Нейронна мережа Хопфілда

Мережа Хопфілда (рис. 1) являє собою нейронну мережу, де всі нейрони пов'язані один з одним. Кожен нейрон отримує на вхід сигнали з усіх інших нейронів і, своєю чергою, розсилає свій сигнал усім іншим нейронам у мережі. Матриця зв'язків такої мережі має бути симетричною, а елементи головної діагоналі мають бути нулями. Ці нулі означають, що сигнал нейрона не передаватиметься йому самому на вхід. Ця умова та умова симетричності повинні забезпечити стійкість системи. У роботі [4] стверджується, що мережа зі зворотними зв'язками є стійкою тільки тоді, якщо виконуються дві згадані вище умови. Стійкість означає, що система через деяку кількість кроків стабілізується і, досягнувши деякого стану, буде надалі залишатися незмінною. В іншому випадку вона може нескінченно коливатися між двома різними станами, не в змозі досягнути рівноваги.

Динаміка мережі Хопфілда, або, інакше кажучи, оновлення стану нейронів, у загальному випадку для окремого нейрона виглядає так [11]:

$$x_i(t+1) = \Phi \left( \sum_{j=1}^n w_{ij} x_j(t) + \theta_i \right), \quad (1)$$

де  $x_i$  – вхідні / вихідні значення нейронів;  
 $w_{ij}$  – вагові коефіцієнти зв'язків між нейронами;  
 $\theta_i$  – зміщення, або порогове значення, яке регулює силу вхідного сигналу.

Функція  $\Phi$  у формулі (1) може бути представлена в різних варіантах (див. табл. 1).

Таблиця 1. Варіанти представлення функції Ф

<p>Для дискретних випадків вона може бути такою:</p> $\text{Sign}(x) = \begin{cases} 1, & x \geq 0 \\ -1, & x < 0 \end{cases}$ $\text{Sign}(x) = \begin{cases} 1, & x \geq 0 \\ -1, & x < 0 \end{cases}$	<p>Для неперервних випадків зазвичай обирається сигмоїдна функція:</p> $g_{\beta}(x) = \frac{1}{1 + e^{-2\beta x}}$ $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$
--	--

Взаємодія нейронів у мережі описується функцією, яка ще називається енергією мережі (2). Головна її властивість полягає в тому, що на кожній ітерації роботи мережі енергія зменшується, причому необов'язково монотонно. Сама нейронна мережа в цьому випадку виступає мінімізатором функції енергії. Тобто після деякого циклу оновлень мережа має досягнути локального мінімуму і після цього перейти в стабільний стан.

$$E(x) = -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n x_i x_j w_{ij} - \sum_{i=1}^n x_i \theta_i. \quad (2)$$

Треба зазначити, що все це справджується тільки в тому випадку, якщо матриця вагових коефіцієнтів мережі є симетричною, тоді функція енергії є функцією Ляпунова.

### Мережа Хопфілда – Танка

У 1985 р. Джон Хопфілд об'єднався з Девом Танком для співпраці з адаптації своєї моделі нейронної мережі до розв'язання оптимізаційних задач [10].

Основна ідея полягала в тому, щоб представити задачу у вигляді функції енергії за умови, що чим кращий розв'язок, тим менша енергія. Тоді можна було би побудувати мережу Хопфілда для цієї функції і спробувати мінімізувати з її допомогою цю функцію [11].

Таким чином, якщо задача оптимізації виглядає так: мінімізувати  $f(x)$  за умови, якщо на неї накладено  $r$  обмежень виду  $C(x) = b$ , то функція енергії для такої задачі може бути описана формулою (3):

$$E(x) = f(x) - \sum_{i=1}^r \alpha_i P_i(x), \quad (3)$$

де  $x$  – вектор вхідних змінних задачі;

$f(x)$  – цільова функція;

$r$  – кількість обмежень;

$\alpha_1, \dots, \alpha_r$  – коефіцієнти, що відображають «відносну важливість» задоволення кожного з обмежень;

$P_i$  – так звані «штрафи», що безпосередньо пов'язані з рівнем задоволеності відповідних обмежень задачі і дорівнюють нулю тільки у випадку, якщо обмеження повністю задоволене. В іншому випадку вони пропорційні задоволеності обмеження. Чим більший рівень задоволення – тим менший коефіцієнт. Допускається, що обмеження в задачі можуть являти собою рівності або нерівності.

Оскільки функція енергії має бути як мінімум квадратичною, то обмеження-рівності в ній подаються в такому вигляді:

$$P_i(x) = (C_i - b_i)^2. \quad (4)$$

Цей вираз дорівнюватиме нулю тільки у випадку, якщо обмеження повністю задоволене, інакше його величина буде пропорційною до ступеня незадоволеності.

Щодо представлення обмежень-нерівностей, то вони мають виглядати таким чином:

$$P_i(x) = \Omega(C_i - b_i). \quad (5)$$

У цій формулі функція омега може бути представлена, наприклад, сигмоїдною функцією.

Треба сказати, що початково підхід зі штрафами розроблявся для розв'язання комбінаторних оптимізаційних задач. Ми ж його розглядаємо в більш вузькому контексті, а саме в контексті розв'язання задач оптимізації обмежень, або, інакше кажучи, для задач задоволення обмежень із гнучкими обмеженнями, де мета полягає в тому, щоб знайти рішення, при якому загальний рівень порушення обмежень буде оптимальним (мінімальним). У таких задачах цільової функції як такої немає, тобто в ній сума штрафів і є цільовою функцією. Тобто першого доданка у формулі (3), по суті, немає.

Тепер постає питання, як, власне, користуватися методом Хопфілда – Танка для розв'язання задач. Після побудови функції енергії наведеним вище способом для неї шукаються похідні по

всіх змінних. Далі ці похідні зіставляються з канонічним виглядом похідної від функції енергії:

$$\Delta E_i(x) = - \left( \sum_{j=1}^n w_{ij} x_j + \theta_j \right) \Delta x_i. \quad (6)$$

Множники, які стоятимуть в отриманих похідних біля  $x_p$  – це будуть значення відповідних вагових коефіцієнтів  $w_{ij}$ , а константні члени являтимуть собою значення  $\theta_j$ .

Найбільше запитань у методі Хопфілда – Танка викликає підбір коефіцієнтів  $a_i$  у формулі  $E(x)$  та ініціалізація стану мережі. Невдалий підбір цих параметрів може суттєво вплинути на подальший хід подій. Наприклад, ініціалізувавши нейрони значеннями, які є можливим розв'язком або є близькими до розв'язку, ми можемо тим самим посприяти тому, що мережа відразу ж потрапить у локальний мінімум. Рекомендується використовувати для ініціалізації нейронів значення  $\sim 0,5$  або ж користуватися підходом [12]. Що ж до вибору коефіцієнтів  $a_p$ , то автори методу не дали жодних рекомендацій стосовно цього, що згодом стало причиною широких дискусій. Про це буде сказано докладніше далі.

Новий метод було успішно застосовано його авторами до розв'язання деяких оптимізаційних задач, серед яких, зокрема, була задача комівояжера. Отримані результати були такими надихаючими, що у 1988 р. Вілсон і Павлев захотіли їх перевірити, підтвердити і поліпшити. Аж тут з'ясувалося, що результати Хопфілда і Танка дійсні тільки для задач відносно невеликого розміру (для задачі комівояжера в них вийшло повторити успішні результати тільки для кількості міст не більше 10), але вдало підібрати коефіцієнти для 900 нейронів набагато важче [10]. Згодом було запропоновано багато методів для покращення розробленої Хопфілдом мережі, але більшість із них реально ненабагато збільшували ефективність методу, тобто покращення були в цілому незначними. Утім, деякі з них заслуговують на увагу.

### Модифікація Вілсона і Павлева мережі Хопфілда – Танка

Як уже було сказано, наріжним каменем методу Хопфілда – Танка є підбір коефіцієнтів для

формули енергії. Вілсон і Павлев запропонували модифікацію цієї формули, де всі ці коефіцієнти замінені одним, адже ймовірність правильно підібрати один коефіцієнт набагато більша, ніж вгадати зі значеннями  $n$  коефіцієнтів. Тож у новій інтерпретації функція енергії набула вигляду [10]:

$$E(x) = f(x) + \frac{\gamma}{2} D(x), \quad (7)$$

де замість коефіцієнтів  $a_i$  з'явився один-єдиний коефіцієнт  $\gamma$ , а  $D(x)$  треба розуміти як відхилення вектора  $x$  (тобто розв'язку) від площини обмежень, заданих для задачі. Тепер якщо задати для  $\gamma$  досить велике значення, то ми гарантуємо знаходження розв'язку.

Докладніше це формулювання функції енергії та сам метод описано в статтях [2] та [10].

### Метод множників Лагранжа для мережі Хопфілда – Танка

Не можна не помітити, що функція енергії за своїм виглядом дуже нагадує функцію Лагранжа. Якщо замість коефіцієнтів  $a_i$  підставити множники Лагранжа, то отримаємо [14]:

$$E(x, \lambda) = f(x) + \sum_{i=1}^r \lambda_i P_i(x). \quad (8)$$

Перевага такого формулювання функції енергії полягає в тому, що коефіцієнти для штрафів  $P_i$  більше не треба підбирати навання. Вони можуть брати участь у нейронній мережі як її елементи нарівні з іншими нейронами, що являють собою змінні задачі.

Формулу їх оновлення можна легко отримати, взявши похідну від  $E$  по  $\lambda_i$ :

$$\frac{\partial E}{\partial \lambda_i} = P_i(x) \quad (9)$$

$$\frac{\partial E}{\partial x_i} = \sum_{j=1}^r \lambda_j \frac{\partial P_j}{\partial x_i}. \quad (10)$$

Автори статей [13; 14] перевірили ефективність свого методу на прикладі задачі про складання розкладу (Job shop scheduling) (табл. 2) і стверджують, що результати роботи їхнього

Таблиця 2. Результати роботи алгоритму LRNN на задачі про складання розкладу

Складність задачі	Операцій	Кількість комп'ютерів	DG (%)	CPU (m:s)
Маленька	6 x 12	5	0,0	0:02
Середня	20 x 200	10	0,4	1:00
Велика	82 x 752	14	6,9	9:34

Примітка. Перша колонка дає оцінку розміру задачі; друга – визначає приблизну кількість операцій, потрібну для розв'язання задачі; третя – кількість задіяних комп'ютерів; четверта – Duality Gap, або різницю між двома кращими розв'язками.

алгоритму значно перевершили результати, отримані для тих самих задач у роботах [18] та [17]. Щоправда, порівняльного аналізу результатів для зазначених методів авторами алгоритму не наведено.

### Машина Больцмана і машина Коші

**Машина Больцмана** базується на асинхронній моделі мережі Хопфілда. Узагалі, мережі Хопфілда поділяються на синхронні та асинхронні залежно від часу передачі сигналу. У синхронній моделі всі нейрони оновлюють свій стан одночасно і оновлені стани кожного з них стають доступними для інших нейронів мережі тільки в наступній ітерації. В асинхронній же мережі це не є вимогою, тобто нейрони оновлюють свій стан врізної, тому може так статися, що два нейрони, які оновлюються одночасно, насправді можуть зробити це з невеликим розривом, у результаті чого на вхід того нейрона, що оновлюється трохи пізніше за сусіда, припаде не старий, а вже оновлений стан іншого нейрона.

Машина Больцмана розширює стандартну асинхронну модель з допомогою методу відпалювання. На кожному кроці довільно обирається один нейрон, для якого обчислюється його оновлений стан, і вираховується  $\Delta E$ . Якщо зміна стану нейрона сприяє зменшенню енергії, то зміна приймається, і стан нейрону оновлюється. Якщо ж ні, то зміна приймається із ймовірністю  $P$ , яка визначається з допомогою критерію Метрополіса [11]:

$$P_i^B(t) = \begin{cases} 1, & \text{if } \Delta E_i(t) < 0 \\ \frac{1}{1 + e^{\frac{\Delta E_i(t)}{T_B(t)}}}, & \text{if } \Delta E_i(t) \geq 0. \end{cases} \quad (11)$$

Для регулювання температури можна використати формулу:

$$T_B(t) = \frac{T_B(t-1)}{1 + t \log(1+r)}, \quad (12)$$

де  $r$  – параметр, який визначає швидкість охолодження.

Великим недоліком цього методу є неможливість його розпаралелення, оскільки оновлення стану нейронів має виконуватись строго послідовно.

**Машина Коші** на противагу машині Больцмана є розширенням синхронної моделі мережі Хопфілда. Тут усі нейрони оновлюються одночасно на кожному кроці. З одного боку, це добре, адже це автоматично робить модель придатною до розпаралелення. З другого боку,

основним недоліком синхронної моделі є те, що нейрони оновлюються неузгоджено, що може призвести до ефекту осциляції, тобто мережа в якийсь момент часу може почати коливатися між двома «стійкими» станами, не в змозі зупинитись на жодному з них, що робить таку модель украй ненадійною. Але модель Коші трохи видозмінила схему оновлення нейронів, що дає змогу великою мірою запобігти таким негативним наслідкам. Якщо позначити вхідне значення нейрона як  $u$ , а вихідне як  $v$ , то функція оновлення стану нейрона виглядатиме як:

$$v_i(t) = \Theta(u_i(t)) \quad (13)$$

$$\Theta(x) = \begin{cases} 1, & x > 0 \\ 0, & x \leq 0 \end{cases} \quad (14)$$

Вхідні значення нейронів оновлюватимуться на кожному кроці, але вихідне значення зможе змінитись тільки після того, як сумарна кількість змін вхідного сигналу за деякий час досягне порога, заданого функцією  $\Theta$ . Таким чином, імовірність того, що два нейрони оновлять свій стан одночасно і незалежно, значно зменшується.

Ще одним покращенням є введення  $P$  – ймовірності оновлення кожного з нейронів у конкретний момент часу, яка визначається розподілом Коші:

$$P_i^C(t) = \begin{cases} s_i(t), & \text{if } v_i(t) = 0 \\ 1 - s_i(t), & \text{if } v_i(t) = 1 \end{cases} \quad (15)$$

$$s_i(t) = P\{v_i(t) = 1\} = \frac{1}{2} + \frac{1}{\pi} \arctan\left(\frac{u_i(t)}{T_C(t)}\right). \quad (16)$$

При цьому температура  $T$  змінюється за «швидкою шкалою»:

$$T_C(t) = \frac{T_0}{1 + \beta t}, \quad (17)$$

де  $T_0$  – початкова температура;

$\beta$  – параметр, що є дійсним числом з проміжкою  $[0, 1]$ , який визначає швидкість охолодження.

Залогом, машина Коші знаходить розв'язки гіршої якості, ніж машина Больцмана, проте в неї все ж залишається перевага, що полягає в можливості розпаралелення обчислень.

**Гібрид машин Больцмана і Коші.** Ця модель є розширенням синхронної моделі мережі Хопфілда, а її стохастичний метод оновлення стану нейронів базується на гібриді правил машини Больцмана і машини Коші і виглядає як:

$$P_i^H(t) = \alpha P_i^C(t) + (1 - \alpha) P_i^B(t). \quad (18)$$

У цій формулі  $\alpha$  є регулюючим параметром і коливається в межах від 0 до 1. Коли його значення 1, то ми отримуємо чисту машину Коші, коли 0 – Больцмана. Для кожної з цих імовірностей вводиться своя температура охолодження.

Експериментальні дослідження показали, що така модель дає результати, ідентичні до результатів машини Больцмана, але гібрид справляється за менший час, що зумовлюється його перевагами розпаралелення, які він успадкував від машини Коші.

### GDS-network

GDS-network – Guarded Discrete Stochastic network (захищена дискретна стохастична мережа) – була розроблена Адорфом і Джонстоном у 1990 р. як альтернатива машині Больцмана. Вона являє собою різновид дискретної стохастичної моделі мережі Хопфілда [8]. Її було розроблено з метою усунути один з критичних недоліків мережі Хопфілда. Річ у тому, що мережа Хопфілда схильна до того, щоб за деяких обставин зупинятися в локальному мінімумі. Це відбувається, коли в мережі лишається кілька активних нейронів, що осцилюють між декількома станами, при цьому нейрони, які могли би змінити ситуацію, є неактивними, оскільки зміна їхнього значення призвела б до збільшення функції енергії. Для того щоб уникнути такого розвитку подій, в основну мережу Хопфілда вводять допоміжну підмережу [9].

Загалом, схема GDS-мережі виглядає таким чином: вхідний шар нейронів представлений у вигляді матриці, де кожен окремий рядок представляє окрему змінну із CSP задачі, а елементи цього рядка – це можливі значення змінної. Теоретично в кожен момент часу тільки один із цих елементів може мати значення 1, інші мають бути 0. Допоміжна підмережа складається з  $N$  нейронів, які є додатковим стовпчиком у матриці нейронів. Вони, як і всі інші нейрони, також можуть набувати тільки значень 0 чи 1. Оновлення стану нейронів відбувається за такою схемою:

- вибираються всі рядки нейронної матриці, де значення елементів (нейронів) є несумісними між собою;
- з усього набору «поганих» рядків обирається довільним чином один;

- для кожного нейрона з обраного рядка, який є несумісним, визначається ступінь його несумісності за формулою:

$$|y_{im} - \eta(x_{im})| |x_{im}|,$$

де  $\{y_{im} : 1 \leq m \leq M\}$  –  $i$ -й рядок матриці нейронів;

$$x_{im} = \sum_{jn} W_{im,jn} y_{jn} + b_{im} - \text{вхід нейрона};$$

$$\eta(x_{im}) = \begin{cases} 1, & x_{im} \geq 0 \\ 0, & x_{im} < 0 \end{cases} - \text{формула, за якою нейрон обраховує свій стан з } x_{im};$$

- з усіх оцінених нейронів обирається найбільш несумісний. Якщо таких декілька, то серед кандидатів обирається довільним чином один, який і робиться сумісним.

Докладніше механізм роботи нейронів допоміжної підмережі описано в [8]. Ініціалізується GDS-network мережа зазвичай таким чином: усім нейронам основної мережі присвоюється значення 0, тоді як усім нейронам допоміжної підмережі – значення 1. Далі поступово деякі з нейронів переключаються в стан 1, і вони або утворюють неконфліктні конфігурації, що є частковими допустимими розв'язками, або, в разі виникнення конфліктів, вдаються до допомоги підмережі, яка змушує конфліктні нейрони змінити свій стан. Таким чином, постійно переключаючись між режимами розширення розв'язку та розв'язання конфліктних ситуацій, мережа прямує до повного розв'язку задачі.

Відмінність такого методу від машини Больцмана полягає в тому, що на кожному кроці вибір нейрона, який буде оновлюватись, є не просто недетермінованим, а ґрунтується на несумісності нейронів.

Такий підхід було застосовано до задачі планування роботи космічного телескопа Хаббла. Як виявилось, планування його роботи є не простою задачею, адже якість його зйомки та доступність тих чи інших космічних об'єктів у полі видимості залежить від багатьох факторів, як-от: низька орбіта, за якою обертається телескоп, та досить невеликий орбітальний період (~95 хвилин). Відповідно, приблизно половину цього часу (~40 хвилин) телескоп є фізично недоступним для зв'язку (тобто тоді, коли він перебуває з другого боку планети). Також є ще декілька «проблемних» зон – зон посиленого випромінювання – на Землі в районі Південної Атлантики, де Хаббл не може нормально функціонувати. Окрім того, бувають деякі проблеми з освітленням та інші

перешкоди, які загороджують собою об'єкти, на які націлений телескоп. Докладніше про це написано в статті [8].

**GENET i Fuzzy GENET**

Алгоритм GENET був представлений світу Вангом і Тсангом у 1991 р. [6] і був логічним продовженням GDS мереж. Як і в GDS, тут нейрони діляться на кластери. Кожен кластер являє собою окрему змінну задачі, а нейрони всередині кластера – можливі значення змінної. Усі нейрони можуть приймати значення 1 або 0, або відповідно «on» або «off». У кожному кластері в кожен момент часу тільки один нейрон може перебувати в стані «on», або відповідно мати значення 1. Фактично це означає, що той нейрон у кластері, який перебуває в стані «on», являє собою значення, яке в цей момент часу присвоєно змінній, яку представляє цей кластер.

Нейрони з різних кластерів пов'язані між собою тільки через змінні, які вони представляють: якщо відповідні змінні зв'язані обмеженнями у задачі, то нейрони також є зв'язаними. Причому в цих кластерах тільки ті нейрони мають зв'язки між собою, які являють собою конфліктні значення змінних. Наприклад, якщо задана задача, як на рис. 2, то мережа GENET виглядатиме так, як на рис. 3.

Вагові коефіцієнти мережі рекомендовано ініціалізувати -1.

Показовим у цьому методі є те, що, на відміну від багатьох інших методів, розробники GENET провели експериментальні дослідження з порівняння ефективності цього методу з ефективностями звичайних методів розв'язання CSP задач. Вони порівняли його з двома алгоритмами локального пошуку: GSAT [19] – на задачі про розфарбування графа, та із min-conflicts hill-climbing (MCHC) – на рандомно-згенерованих CSP задачах.

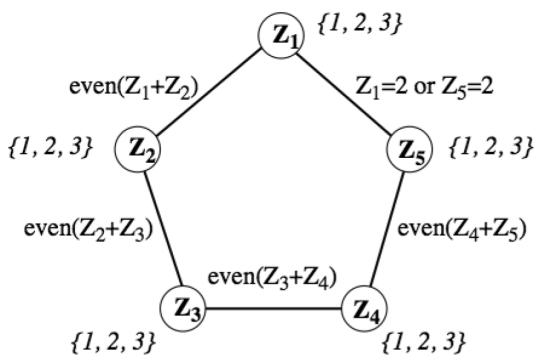


Рис. 2. Приклад CSP задачі

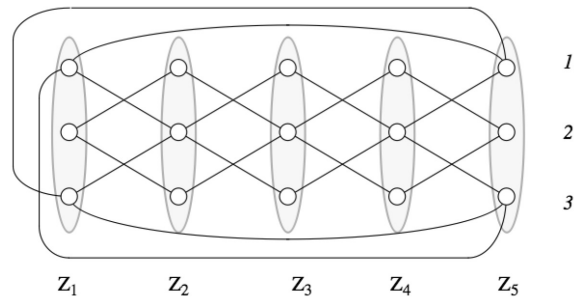


Рис. 3. Приклад мережі GENET для задачі з рис. 2

У результаті першого порівняння було виявлено, що обидва алгоритми справилися з усіма тестовими задачами. Нижче (табл. 3, 4) наведено результати в цифрах [6].

Таблиця 3. Результати роботи алгоритму GSAT на складних задачах розфарбування графа

Граф		Середній час	Середня кількість кроків
Вершина	Кольори		
125	17	8,0 год	65 197 415
125	18	30 с	65 969
250	15	5,0 с	2 839
250	29	1,8 год	7 429 308

Таблиця 4. Результати роботи алгоритму GENET на складних задачах розфарбування графа

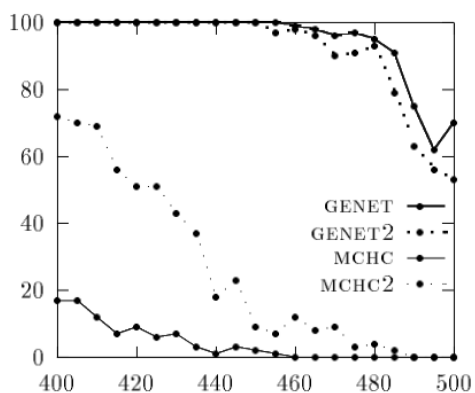
Граф		Середній час	Середня кількість кроків
Вершина	Кольори		
125	17	2,6 год	1 626 861
125	18	23 с	7 011
250	15	4,2 с	580
250	29	1,1 год	571 748

Другий експеримент, у ході якого перевірялася ефективність GENET та деякої його модифікації GENET2, а також алгоритму MCHC і його модифікації MCHC2, показав, що GENET знову виявився переможцем, причому з разючим відривом. Тестовими задачами виступали CSP задачі з 50 змінними, кожна з яких мала домен з 10 значень. Обмеження для цих задач генерувалися рандомно. На графіку (рис. 4) можна побачити успішність кожного з алгоритмів, яка вимірюється у відсотках розв'язаних задач від загальної кількості задач [6].

**Fuzzy GENET.** Логічним продовженням GENET стала її версія для CSP задач із гнучкими обмеженнями, тобто для Fuzzy CSP (FCSP). Так з'явився алгоритм під назвою Fuzzy GENET [20].

Відмінність алгоритму Fuzzy GENET від базового GENET полягає в тому, що в ньому кожен зв'язок між нейронами в нейронній мережі описується двійкою: рівнем задоволення обмеження, пов'язаного з цими змінними (всі обмеження в задачі бінарні), та ваговим коефіцієнтом  $W_{i_y j_z}$ . Вагові коефіцієнти ініціалізуються як  $W_{i_y j_z} = \alpha_{i_y j_z} - 1$ . Усе інше, в принципі, так само, як у GENET.

Автори статті про Fuzzy GENET також провели низку експериментів для порівняння ефективностей GENET та Fuzzy GENET на звичайних CSP задачах. Обидва алгоритми показали схожі результати, що означає, що Fuzzy GENET є універсальним і загалом є гідною заміною своєму попереднику. Нижче наведено порівняльну таблицю (табл. 5) двох алгоритмів для задачі про ферзів.



**Рис. 4.** Порівняння результатів роботи алгоритмів GENET, GENET2, MCHC та MCHC2 на випадкових CSP задачах. На горизонтальній осі – кількість обмежень у задачі; на вертикальній – успішність кожного алгоритму у % розв'язаних задач

**Таблиця 5.** Порівняння результатів роботи алгоритмів GENET та Fuzzy GENET на задачі про ферзів, де  $n$  – кількість ферзів у задачі

$n$	GENET	Fuzzy GENET
10	0,06 с	0,08 с
20	0,48 с	0,52 с
30	2,09 с	2,18 с
40	7,27 с	9,52 с
50	12,49 с	13,03 с
60	23,04 с	23,13 с
70	32,03 с	30,58 с

## Висновки

Джон Хопфілд був першопрохідцем у намаганні адаптувати свою нейронну мережу до розв'язання CSP задач. Незважаючи на невдалі перші спроби та критику, пов'язану з тим фактом, що опубліковані ним результати не

підтвердилися під час перевірки їх іншими дослідниками, все ж Хопфілд своєю роботою заклав фундамент для подальших досліджень цієї теми. Мережа Хопфілда – Танка добре висвітлює всі проблемні місця в методиці застосування мережі Хопфілда до розв'язання CSP задач і тим самим стимулювала активний розвиток досліджень, спрямованих на покращення цих недоліків.

Першим серйозним покращенням мережі було введення нової формули енергії Вілсоном і Павлевим [10], де всі коефіцієнти було замінено одним  $\gamma$ . Це значно спростило ініціалізацію нейронної мережі, яка потребувала підбору «чарівних» значень для великої кількості вхідних параметрів (коефіцієнтів обмежень, вагових коефіцієнтів, нейронів), адже від початкової конфігурації значно залежало, як далі розвиватиметься мережа, чи не потрапить вона в локальний мінімум, чи не стане безкінечно осцилювати між двома станами. Послідовником Вілсона та Павлева став Лах [14], який побачив у функції енергії мережі Хопфілда схожість із функцією Лагранжа. Він же був першим, хто порівнює результати своїх експериментальних досліджень не тільки з попередниками, але й із зовсім іншими алгоритмами розв'язання цієї ж задачі, і результати порівняння говорили на користь Лаха.

Наступним етапом покращення мережі Хопфілда стала її гібридизація з метаевристичним алгоритмом – методом відпалювання. Так з'явилися машина Больцмана, машина Коші та гібрид цих двох методів [11]. Загалом, на думку автора статті [10], гібридизація нейронних мереж із метаевристичними алгоритмами – це один із двох перспективних напрямів розвитку методів на основі нейронних мереж у сфері розв'язання CSP задач. Другим перспективним напрямом він вважає пошук альтернативних підходів у сфері нейронних мереж, тобто застосування інших типів мереж для розв'язання CSP або ж розробка інших моделей проектування CSP задач на нейронній мережі.

Першим справді серйозним нащадком мережі Хопфілда стала GDS-мережа, яка була спроектована для розв'язання задачі планування розкладу телескопа Хаббла [8]. А її логічні продовження – алгоритми GENET [6] та Fuzzy GENET [20] – стали першою спробою застосувати мережу Хопфілда до розв'язання FCSP задач.

З усього сказаного вище можна зробити висновок, що нейронні мережі, зокрема нейронні мережі Хопфілда, незважаючи на зіпсований авторитет, успішно застосовуються до розв'язання CSP задач. Причому різноманіття методів, що



базуються на нейронній мережі Хопфілда, досить велике. Щоправда, розвиток цих методів у напрямку застосування їх до розв'язання FCSP задач чомусь не пішов далі одного алгоритму, що, можливо, зумовлено обмеженими можливостями

мережі. Адже зі структури Fuzzy GENET стає очевидним те, що вона не застосовна до FCSP задач, де змінні мають неперервні домени. Тож, можливо, варто шукати вихід у розробці та модифікації інших нейронних мереж.

#### Список літератури

1. Галковська Л. О. Алгоритми розв'язання розподіленої задачі задоволення обмежень / Л. О. Галковська // Наукові записки НАУКМА. – 2013. – Т. 151 : Комп'ютерні науки. – С. 139–148.
2. Aiyer S. V. B. Solving Combinatorial Optimization Problems Using Neural Networks : Technical Report CUED/FINFENG/TR 89 / S. V. B. Aiyer ; Cambridge University Engineering Department. – Cambridge, U.K., 1991.
3. Brito I. Synchronous, Asynchronous and Hybrid Algorithms for DisCSP / I. Brito // Proceedings of the Tenth International Conference on Principles and Practice of Constraint Programming. Lecture Notes in Computer Science. – 2004. – Vol. 3258. – P. 791.
4. Cohen M. A. Absolute stability of global pattern formation and parallel memory storage by competitive neural networks / M. A. Cohen, S. G. Grossberg // IEEE Transactions on Systems, Man and Cybernetics. – 1983. – Vol. 13. – P. 815–826.
5. Dubois D. The calculus of fuzzy restrictions as a basis for flexible constraint satisfaction / D. Dubois, H. Fargier, H. Prade // 2nd IEEE International Conference on Fuzzy Systems San Francisco (California, USA). – 1993. – Vol. 2. – P. 1131–1136.
6. GENET: A connectionist architecture for solving constraint satisfaction problems by iterative improvement / A. Davenport, E. Tsang, C. J. Wang, K. Zhu // Proceedings of AAAI. – 1994. – P. 1–6.
7. Hermann M. On the Complexity of Fuzzy Boolean Constraint Satisfaction Problems with Applications to Intelligent Digital Photography / M. Hermann, F. Richoux // Circuits, Communications and System (PACCS) – Third Pacific-Asia Conference. – 2011. – P. 1–4.
8. Johnston M. D. Scheduling with Neural Networks the Case of Hubble Space Telescope / M. D. Johnston, H. M. Adorf // Computer and Operations Research. – 1992. – Vol. 19. – P. 209–240.
9. Johnston M. D. A discrete stochastic neural network algorithm for constraint satisfaction problems / M. D. Johnston, H. M. Adorf // IJCNN International Joint Conference. – 1990. – P. 1–10.
10. Kate A. S. Neural Networks for Combinatorial Optimization: A Review of More Than a Decade of Research / A. S. Kate // Journal on Computing. – 1999. – Vol. 11, № 1. – P. 15–34.
11. Lagoudakis M. G. Neural Networks and Optimization Problems. A Case Study: The Minimum Cost Spare Allocation Problem / M. G. Lagoudakis // The Center for Advanced Computer Studies – University of Southwestern Louisiana. – 1997. – P. 1–15.
12. Lai W. K. Initialising the Continuous Hopfield Net / W. K. Lai, G. G. Coghill // Proceedings International Conference on Neural Net. – 1994. – Vol. 7. – P. 4640–4644.
13. Luh P. B. Lagrangian relaxation neural network for unit commitment / P. B. Luh // Department of Electronics and System Engineering – Connecticut University, Storrs, CT, USA. – 1999. – P. 1–6.
14. Luh P. B. Lagrangian Relaxation Neural Networks for Job Shop Scheduling / P. B. Luh, X. Zhao, Y. Wang // Proceedings of International Conference on Robotic and Automation, Leuven, Belgium. – 1998. – P. 1799–1804.
15. Osman I. H. Metaheuristics: A Bibliography / I. H. Osman, G. Laporte // Annals of Operations Research. – 1996. – P. 513–623.
16. Ruttkay Z. Fuzzy constraint satisfaction / Z. Ruttkay // 3rd IEEE Conference on Fuzzy Systems, Orlando (Florida, USA). – 1994. – Vol. 2. – P. 1263–1268.
17. Sabuncuoglu I. A Neural Network Model for Scheduling Problems / I. Sabuncuoglu, B. Gurgun // European Journal of Operational Research. – 1996. – Vol. 9, № 2. – P. 288–299.
18. Satake K. Neural Network Approach for Minimizing the Makespan of the General Job-shop / K. Satake, K. Morikawa, N. Nakamura // International Journal of Production Economics. – 1994. – Vol. 33, № 1. – P. 67–74.
19. Selman B. Domain-Independent Extensions to GSAT: Solving Large Structured Satisfiability Problems / B. Selman, H. Kautz // Proceedings of the 13th international joint conference on Artificial intelligence. – 1993. – Vol. 1. – P. 290–295.
20. Wong J. H. Extending Genet to Solve Fuzzy Constraint Satisfaction Problems / J. H. Wong, H. Leun // AAAI Proceedings. – 1998. – P. 1–6.
21. Zadeh L. A. Fuzzy sets / L. A. Zadeh // Information and Control. – 1965. – Vol. 8, № 3. – P. 338–353.

*L. Galkovska*

## APPLYING HOPFIELD NETWORK TO CONSTRAINT SATISFACTION PROBLEMS

*This work introduces review of the existing methods for solving CSP and FCSP, which are based on Hopfield network. The first attempt to apply this type of neural networks to solving CSP was made by Hopfield himself. This new algorithm was followed by the sequence of its modifications developed by others. So, all of the algorithms described in the current article are modifications of the single algorithm, which are extending and supplement one another. Also the article contains some comparative characteristics of the described methods with other existing methods for solving CSP (which are not based on neural network).*

**Keywords:** CSP, Constraint Satisfaction Problem, neural network, Hopfield network.

*Матеріал надійшов 21.05.2015*