

Zeitschrift: Bulletin des Schweizerischen Elektrotechnischen Vereins, des Verbandes Schweizerischer Elektrizitätsunternehmen = Bulletin de l'Association suisse des électriciens, de l'Association des entreprises électriques suisses

Herausgeber: Schweizerischer Elektrotechnischer Verein ; Verband Schweizerischer Elektrizitätsunternehmen

Band: 84 (1993)

Heft: 25

Artikel: Programmieren mit Randbedingungen : eine neue Methode für komplexe Planungs- und Optimierungsprobleme

Autor: Lebsanft, Ernst

DOI: <https://doi.org/10.5169/seals-902766>

Nutzungsbedingungen

Die ETH-Bibliothek ist die Anbieterin der digitalisierten Zeitschriften auf E-Periodica. Sie besitzt keine Urheberrechte an den Zeitschriften und ist nicht verantwortlich für deren Inhalte. Die Rechte liegen in der Regel bei den Herausgebern beziehungsweise den externen Rechteinhabern. Das Veröffentlichen von Bildern in Print- und Online-Publikationen sowie auf Social Media-Kanälen oder Webseiten ist nur mit vorheriger Genehmigung der Rechteinhaber erlaubt. [Mehr erfahren](#)

Conditions d'utilisation

L'ETH Library est le fournisseur des revues numérisées. Elle ne détient aucun droit d'auteur sur les revues et n'est pas responsable de leur contenu. En règle générale, les droits sont détenus par les éditeurs ou les détenteurs de droits externes. La reproduction d'images dans des publications imprimées ou en ligne ainsi que sur des canaux de médias sociaux ou des sites web n'est autorisée qu'avec l'accord préalable des détenteurs des droits. [En savoir plus](#)

Terms of use

The ETH Library is the provider of the digitised journals. It does not own any copyrights to the journals and is not responsible for their content. The rights usually lie with the publishers or the external rights holders. Publishing images in print and online publications, as well as on social media channels or websites, is only permitted with the prior consent of the rights holders. [Find out more](#)

Download PDF: 15.04.2026

ETH-Bibliothek Zürich, E-Periodica, <https://www.e-periodica.ch>

Constraint Programming (CP), eine neuere Entwicklung der Informatik, macht es möglich, benutzerfreundliche Applikationen für komplexe Aufgaben wie zum Beispiel Auslastungsoptimierung von Netzen, Personaleinsatzplanung, Produktionsplanung oder Fahrplangenerierung schnell und kostengünstig zu realisieren. Der Beitrag befasst sich mit der Funktionsweise des Constraint Programming, seinen Vorteilen gegenüber anderen Ansätzen sowie mit Beispielen von operationellen Systemen und dem ökonomischen Potential von CP.

Programmieren mit Randbedingungen – eine neue Methode für komplexe Planungs- und Optimierungsprobleme

■ Ernst Lebsanft

Angesichts einer hartnäckigen Rezession und einer steigenden Wettbewerbsintensität erhält das Stichwort «lean» für viele Betriebe in nahezu allen Bereichen zunehmend an Bedeutung. Eigentlich steckt nichts anderes dahinter als ein sparsamerer Umgang mit vorhandenen Ressourcen wie Lagerplatz, Maschinen, Anlagen, Material, Rohstoffe, Zeit und nicht zuletzt auch Mitarbeiter. Probleme der Ressourcenallokation, der Produktionskontrolle, der Reihenfolgeplanung (Scheduling) und der Planung (Planning)¹ bekommen strategische Bedeutung. Typische Beispiele für solche Aufgaben sind:

- Netzplanung in Telekommunikations- und Energienetzen
- Optimierung der Einsatzplanung von Mitarbeitern in Verkehrsbetrieben, Produktionsstätten, Krankenhäusern unter Einhaltung oft komplexer Randbedingungen
- Reihenfolgeplanung von Produktionsaufträgen in Nahrungsmittel- und Pharmabetrieben

¹ Planung bedeutet vor allem die Ermittlung und adäquate Aneinanderreihung von Aufgaben zwecks Erreichens eines vorgegebenen Ziels, während unter Scheduling das Auffinden einer Reihenfolge für vorgegebene Aufgaben zu verstehen ist, wobei diese Reihenfolge ebenfalls vorgegebenen Bedingungen wie zum Beispiel Optimalität hinsichtlich irgendeines Kriteriums zu genügen hat.

- Routenplanung für Verkehrsmittel und Transportgüter
- Arbeitsplangenerierung in Fertigungsbetrieben
- Auslastungsoptimierung von Bahnhöfen

Besonders interessant und anspruchsvoll sind die zu lösenden Aufgaben, wenn man aufgrund äußerer Einflüsse (u.a. Störungen) von einem bestehenden, meistens gut ausgefüllten Plan abweichen und unter den neu eingetretenen, erschwerenden Bedingungen und unter Termindruck eine konfliktfreie und optimale Lösung finden muss. In solchen Ausnahmesituationen ist man ohne rechnergestützte Systeme meist verloren.

Warum Constraint Programming?

Probleme, wie sie oben geschildert wurden, lassen sich formal wie folgt darstellen:

Gegeben:

- eine endliche Menge von Objekten wie zum Beispiel Aufgaben, Maschinen, Mitarbeiter usw.
- eine Menge von Randbedingungen auf diesen Objekten wie zum Beispiel Verfügbarkeiten, Unverträglichkeiten, Vorgänger-Nachfolger-Beziehungen usw.

Zu finden ist eine Lösung, die

- alle Randbedingungen einhält,

Adresse des Autors:

Dr. Ernst Lebsanft, Synlogic AG,
Rottmannsbodenstrasse 300, 4102 Binningen.

- möglichst irgendeine ökonomische Funktion wie zum Beispiel Termineinhaltung, Gesamtkosten, Auslastung usw., optimiert

So einfach sich diese Probleme beschreiben lassen, so schwer sind sie wegen der kombinatorischen Explosion zu lösen. Ferner ändern sich die Objekte und Randbedingungen relativ schnell im Laufe der Zeit. Brauchbare Lösungen müssen daher derartige Änderungen leicht und effizient handhaben können.

Randbedingungsprobleme waren und sind ein typischer Gegenstand des Operations Research (OR). Es hat sich jedoch herausgestellt, dass auf OR-Ansätzen beruhende konventionelle Softwarelösungen oft lange Entwicklungszeiten hatten und oft immer noch haben, schwerfällig und schwer zu pflegen sind. Der konventionelle Ansatz, nämlich problemspezifische Programme in einer prozeduralen Sprache zu erstellen, kann die heutigen praktischen Anforderungen nicht flexibel genug erfüllen.

Es ist daher nicht verwunderlich, dass versucht wurde, solche Probleme mit anderen Technologien, wie zum Beispiel mit wissensbasierten Systemen (WBS) und den WBS verwandten Technologien [1; 2] zu lösen, die bereits mehrfach im Bulletin SEV/VSE² thematisiert wurden und darum an dieser Stelle nicht weiterer Erörterung bedürfen. In der Praxis finden sich einige auch ökonomisch erfolgreiche WBS, zum Beispiel [3; 4]) sowie gelegentlich gemischttechnologische Lösungen wie zum Beispiel die wissensbasierte gesteuerte Suche in grossen Suchräumen mit Algorithmen wie A* zur Lösung von Reihenfolgeproblemen [5]. Im Falle realer Randbedingungsprobleme ist jedoch auch die Technologie wissensbasierter Systeme nur begrenzt tauglich.

In jüngster Zeit ist mit dem Constraint Programming (CP) in seinen verschiedenen Spielarten³ nicht nur ein sehr vielversprechender Lösungsansatz mit ersten Nachweisen der Praxistauglichkeit [6] entstanden, sondern vor allem auch Software-Werkzeuge, mit denen sich diese Technologie effizient anwenden lässt. In diversen Applikationen konnte gezeigt werden, dass CP verschiedenen bisher angewandten Techniken, wie zum Beispiel auch der Linearen Programmierung, deutlich überlegen ist. Im folgenden wird versucht, einige grundlegende Ideen des Constraint Programming⁴, beispielhafte Anwendungen und das Potential dieser Technologie vorzustellen.

² Zum Beispiel schwerpunktmässig in den Heften 15/1988 und 3/1989.

³ Constraint-Based Programming (CBP), Constraint Satisfaction Programming (CSP), Constraint Logic Programming (CLP).

⁴ Die folgenden Ausführungen orientieren sich an den Möglichkeiten des Werkzeugs Charme. Die gesamte Vielfalt des CP darzustellen, würde ein Buch füllen.

Elemente des Constraint Programming

Wesentlich am Constraint Programming ist ein neuer Ansatz der Modellierung eines Sachverhalts, das heisst eine neue Sichtweise auf ein gegebenes Problem. Traditionelle Programmierung modelliert einen realweltlichen Ausschnitt prozedural, das heisst mit den Konstruktiven Sequenz, Verzweigung und Schleife. Der objektorientierte Denkansatz benutzt die Konstrukte Objekt (Klasse), Nachricht (zum Austausch von Information zwischen Objekten) und Methode (zu einem Objekt gehörig). Es gibt zahlreiche weitere Modellierungs- und damit Programmieransätze⁵ wie den logikorientierten, den regelorientierten, den relationsorientierten, den planorientierten oder den operatororientierten Ansatz, alle einschliesslich der heute favorisierten objektorientierten Programmierung von der Informatik-Teildisziplin Künstliche Intelligenz seit Jahrzehnten entwickelt [7].

Constraint Programming modelliert ein Problem als eine Menge von Randbedingungen, die auf Variable angewandt werden. Die Problemlösung besteht aus den Variablenwerten, die alle Randbedingungen gemeinsam erfüllen. Vereinfacht gesagt, besteht CP in der Formulierung der Randbedingungen, der Definition der Variablen und der Festlegung der Wertebereiche⁶. Die Variablen repräsentieren die Unbekannten des Problems und die Constraints die Restriktionen, denen die möglichen Werte der Variablen innerhalb einer Lösung unterworfen sind. Der initiale Suchraum (Ausgangssuchraum) ist also das kartesische Produkt aller Variablenomänen (d.h. alle Kombinationen aller Variablen mit allen Werten).

Das Auffinden einer Lösung ist also im wesentlichen ein Suchproblem. Die grundlegende Idee des CP ist die fundamentale Reduktion des initialen Suchraumes, indem durch dynamische, wiederholte Anwendung der Constraints laufend Variablenwerte eliminiert werden, die nicht in einer möglichen Lösung vorkommen können. Dieser Prozess – Constraint Propagation genannt – wird solange fortgesetzt, bis alle Variablen einen Wert angenommen haben. Die Gesamtheit dieser Werte stellt die Lösung dar. Constraints verknüpfen Variable miteinander. Wenn die Domäne einer Variablen verändert wird, wird diese Information zu den anderen Variablen, die durch die Randbedingung assoziiert sind, propagiert. Wenn dann die Domäne einer der Variablen reduziert ist, wird der Prozess fortgesetzt. Passiert es dann zum Beispiel, dass eine Domäne leer wird, kann dies keine Lösung sein, der Prozess muss zurückgesetzt bis zu dem Punkt werden, an

⁵ Von H. Stoyan auch «Programmierstile» genannt.

⁶ Auch «Domäne» einer Variable genannt.

dem für eine Variable eine Wertwahl vorgenommen wurde (Backtracking), und es muss von dort an mit einem neu zu wählenden Wert propagiert werden.

Probleme, die ohne Backtracking gelöst werden können, nennt man deterministisch. Bei nichtdeterministischen Problemen werden im Laufe des Propagationsprozesses getroffene Wertselektionen solange neu getroffen, bis eine Lösung erreicht ist oder bis feststeht, dass es keine gibt. Besonders interessant am CP-Ansatz ist, dass man im letzteren Fall durch Constraint Relaxation, das heisst dem «Aufweichen» an sich harter Randbedingungen zur Laufzeit, mindestens zu Näherungslösungen gelangen kann [8]. Bei Terminierungsaufgaben (z.B. Stundenpläne, Fahrpläne) kommt es öfter vor, dass es an sich keine Lösung gibt, die alle Randbedingungen erfüllt. Constraint Propagation und Backtracking werden nachfolgend durch ein berühmtes und zugleich leicht zu verstehendes Beispiel illustriert.

Das Acht-Damen-Problem

Die Aufgabe lautet, acht Damen auf einem Schachbrett so zu positionieren, dass sie sich gegenseitig nicht schlagen können, dass also folgende Constraints erfüllt sind:

- alle Damen sind auf verschiedenen Horizontalen
- alle Damen sind auf verschiedenen Vertikalen
- alle Damen sind auf verschiedenen Diagonalen

Man startet den Prozess mit der Positionierung einer Dame (in den Bildern durch das Herz-Symbol dargestellt) auf einem beliebigen Feld, worauf durch Constraint Propagation sofort alle verbotenen Felder gelöscht (grau im Bild) werden (Bild 1). Durch aufeinanderfolgende Wahl der Felder B2, C4 und D1 für die zweite, dritte und vierte Dame wird der Zustand in Bild 2 erreicht.

Die Wahl des Feldes D1 für die vierte Dame führt zum Löschen der dunkelgrauen Felder durch Constraint Propagation. In den Reihen E–H können dann durch die weitere Propagation der Randbedingungen nur noch die blauen Felder E8, F5, G3 und H6 besetzt werden (in dieser Reihenfolge, Bild 3). Dieses Ergebnis wurde durch die ersten vier Selektionen (Reihen A–D) zwangsläufig erreicht.

Man nehme nun an, die vierte Wahl wäre nicht D1, sondern D6 gewesen. Constraint Propagation führt dann zu den einzig möglichen Besetzungen G5 und F3 in den Reihen G und F, während in Reihe H kein Feld mehr frei ist und in Reihe E dafür zwei (Bild 4), das heisst es wird keine Lösung erreicht.

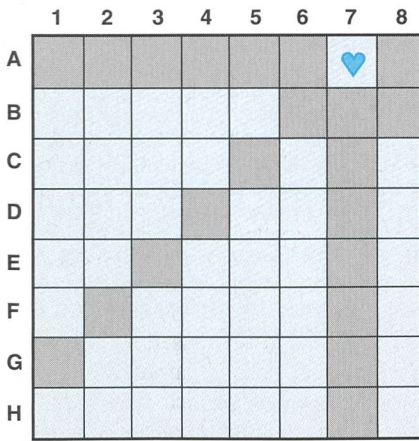


Bild 1 Das Acht-Damen-Problem (1)

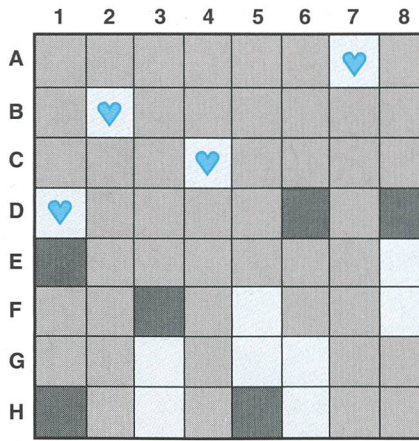


Bild 2 Das Acht-Damen-Problem (2)

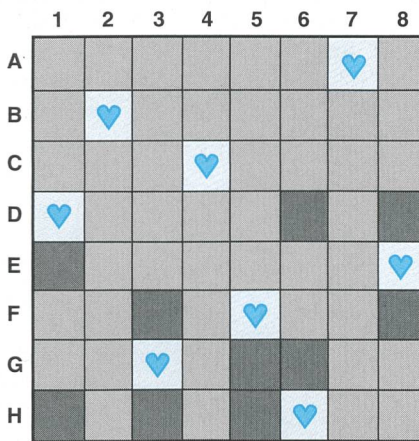


Bild 3 Das Acht-Damen-Problem (3)

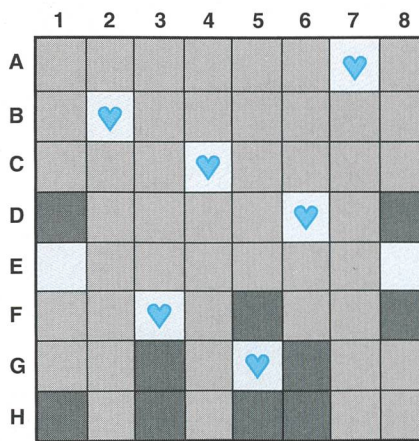


Bild 4 Das Acht-Damen-Problem (4)

Jetzt ist Backtracking notwendig, das heisst das Zurücksetzen des ganzen Systems bis zur letzten Wahl (d.h. D6 für die vierte Dame), die zum Fehlschlag führte, um dort mit einer neuen Wahl den Propagationsprozess wieder aufzusetzen. Eine Wahl von zum Beispiel D8 führt dann wieder zu einem Konflikt, der durch erneutes Backtracking schliesslich zur Wahl von D1 und zur Lösung führt, wie der Leser leicht selbst nachprüfen kann. Sollte zum Beispiel auch dann keine Lösung zu finden sein, muss man noch einen Schritt weiter zurücksetzen usw.

Aus dem Beispiel dürfte klar geworden sein, dass bei constraintbasierten Systemen der Selektionsstrategie für die Variablen eine entscheidende Bedeutung dafür zukommt, wie schnell eine Lösung gefunden wird.

Das beschriebene Beispiel kann natürlich auch mit anderen Ansätzen gelöst werden. Bis 1991 war es mit den besten Suchalgorithmen jedoch nicht möglich, das N -Damen-Problem für $N > 1000$ zu lösen. Mit einem CSP-Ansatz ist es 1991 gelungen, das Problem für $N = 1000000$ auf einer Sparc Station 1 in einigen Minuten zu lösen [9], was die Mächtigkeit dieses Ansatzes für derartige Aufgabenstellungen demonstriert.

Constraintbasierte Applikationen

Constraint Programming ist bereits bei zahlreichen komplexen Planungs- und Optimierungsproblemen erfolgreich verwendet worden. Ein Beispiel ist ein italienisches Entscheidungsunterstützungssystem für das Verkehrsmanagement in einem Telefonnetzwerk zur Aufrechterhaltung der Netzperformance im Falle von Fehlern oder Überlastungen [10]. Das System unterstützt speziell das sogenannte Temporary Alternative Routing. Gleichartige Aufgaben gibt es in jeder Art von Netzwerken, in Datennetzen ebenso wie in Energie- und Verkehrsnetzen.

Im Bereich der Produktionsplanung und -steuerung sind zahlreiche Systeme unter Einsatz von CP realisiert worden, und zwar

sowohl für Einzelfertiger als auch für Klein- und Grossserienfertiger [11]. Weitere bekanntgewordene Anwendungen sind etwa interaktive Systeme für die Einsatzplanung von Transportmitteln und Personal bei Verkehrsbetrieben, die dann auch bei auftretenden Störungen die Umdisposition unterstützen. Folgende Anwendungen seien noch kurz zitiert [12]:

- Minimierung zu fahrender Zugkilometer bei gegebenem Fahrplan bei den holländischen Eisenbahnen
- Minimierung von Leerkilometern von Lokomotiven bei der SNCF
- Minimierung der Schiffsbewegungen im Hafen von Singapur
- Optimierung von Stabilität und Trimmung bei der Beladung von Rollon-Rolloff-Containerschiffen im Hafen von La Spezia/Italien
- Produktionsplanung in einer Ölfabrik von Elf (Frankreich)
- Optimierung der Montage von Turbinenteilen bei Sochata (Frankreich)
- kostenoptimierende Personaleinsatzplanung bei den Verkehrsbetrieben von Poitiers/Frankreich
- Einsatzplanung mobiler Einheiten bei der französischen Nationalgarde
- Optimierung des Ressourceneinsatzes beim Sicherheitsmanagement der Olympischen Winterspiele 1992 (Frankreich)

Durch eigene Untersuchungen für Unternehmungen unterschiedlicher Branchen (z.B. Baustoffe, Lebensmittel) konnte der Autor ferner feststellen, dass CP bisher nicht denkbare Planungssysteme ermöglicht, mit denen sich erhebliche Einsparungen in kurzer Zeit erzielen lassen. CP ermöglicht Lösungen von Aufgaben, an denen das Operations Research bisher gescheitert ist. Wichtig ist auch die Feststellung, dass bei den realisierten Anwendungen CP meist mit anderen Techniken erfolgreich integriert wurde.

Programmierwerkzeuge für CP

Inzwischen bietet der Markt eine Reihe von Software Tools für die Programmierung mit Randbedingungen, so zum Beispiel die in Tabelle I beschriebenen [13] an. Die verschiedenen Werkzeuge weisen sehr unterschiedliche Leistungsprofile auf, so dass eine

Tool	Plattform	Kommentar
Ilog Solver	Diverse	Scheduling, Konfiguration, Planung
Charme	Unix	Diskrete kombinatorische Probleme
Chip	Diverse	Diskrete kombinatorische Probleme, propositionale Logik, LP
Prolog III	Diverse	Prolog erweitert um CP

Tabelle I

professionelle Evaluation auf Basis einer präzisen Anforderungsspezifikation notwendig ist. Generell lässt sich sagen, dass nahezu alle CP-Tools auf moderne Art und Weise realisiert wurden und eine hohe Integrationsfähigkeit aufweisen.

Ausblick

Programmieren mit Randbedingungen fristet in der Schweiz im Unterschied zu anderen europäischen Ländern trotz des hohen ökonomischen Potentials⁷ noch ein Mauerblümchen-Dasein. Von Aktivitäten an Hochschulen einmal abgesehen, sind dem Autor keine operationellen industriellen Anwendungen bekannt, obwohl es mit Sicherheit zahlreiche Aufgaben gibt, die man mit CP besser als bisher oder überhaupt erst lösen könnte, beispielsweise in der Konsumgüterindustrie, der Pharmaindustrie, im Maschinenbau, in städtischen Verkehrsbetrieben, bei den SBB, beim Militär oder im Telekommunikationssektor. Vielleicht ist es ein Anreiz, dass die meisten im Ausland bekanntgewordenen Anwendungen einen erheblichen Return on Investment hatten.

Literatur

[1] Ernst Lebsanft, André Prim und François Braun: Aus alten Fehlern lernen – Case-Based Reasoning, Bulletin SEV/VSE 83(1992)17, S. 17–22.

[2] Ernst Lebsanft und Hans Stienen: Höhere Effizienz an der Verkaufsfond – Computerunterstützung im Verkauf, Bulletin SEV/VSE 84(1993)11, S. 11–25.

⁷ Es gibt Bereiche, in denen ohne weiteres jährliche Einsparungen von Fr. 1 000 000 und mehr mit einer Investitionssumme von Fr. 1 000 000 oder auch deutlich weniger bei einer Projektlaufzeit von einem guten Jahr erzielt werden können.

[3] C. Mossiadis: Arbeitsplangenerierung für die Fertigung von Kleinblechteilen. In: T. Messe, H. Winklhofer (Hrsg.): Beiträge zum 6. Workshop Planen und Konfigurieren, Bayerisches Forschungszentrum für wissensbasierte Systeme (Forwiss), Erlangen, März 1992, S. 102–111.

[4] F. Bajus: Rationalisierungseffekte durch das Arbeitsplangenerierungssystem APGEN bei der Deutschen Aerospace Airbus GmbH. Vortrag anlässlich des Workshops «Wissensbasierte Systeme in Konstruktion und Produktion» der Insiders GmbH am 11. März 1993 in Hamm.

[5] F. Braun und E. Lebsanft: Système d'ordonnement expert de lignes robotisées d'emballage pour l'industrie alimentaire. In: EC2 (Ed.): Tenth International Workshop Expert Systems and their Applications, Specialized Conference AI, Food Processing, Biotechnological, Chemical and Pharmaceutical Industries, Avignon May 28 – June 1st, 1990, Nanterre, France, pp. 263–274.

[6] Ernst Lebsanft: Wissensbasierte Systeme in der Logistik. Logistik Spektrum, S. 13–15, Wirtschaftssupplement zu f+h 4, April 1993 sowie Distribution 4, April 1993.

[7] Herbert Stoyan: Künstliche Intelligenz – Sprachen und Systeme. Computer Magazin 3/87, S. 68–71.

[8] Joachim Hertzberg, Hans-Werner Gúsgen, Angelika Voss, Manfred Fidelak, Hans Voss: Relaxing

constraint networks to resolve inconsistencies. In: Joachim Hertzberg und Andreas Günter (Hrsg.). Beiträge zum 2. Workshop Planen und Konfigurieren, Arbeitspapiere der GMD 310, St. Augustin, 1988, S. 91–101.

[9] Jacqueline Chabrier, Jean-Jacques Chabrier et François Trousset: Résolution Efficace d'un Problème de Satisfaction de Contraintes; Le Million de Reines. In: EC2 (Ed.): Eleventh International Conference Expert Systems and Their Applications, General Conference Tools, Techniques and Methods, Vol. 1, Avignon, May 27–31, 1991, Nanterre, France, pp. 261–268.

[10] Giovanni Martini, Corrado Moiso and Marisa Porta: A Constraint Logic Programming Based Decision Support System for Network Traffic Management. In: EC2 (Ed.): Twelfth International Conference Artificial Intelligence, Expert Systems, Natural Language, Specialized Conferences, Volume 2, Avignon, June 1–6, 1992, Nanterre, France, pp. 381–390.

[11] H. Atabakhsh: A survey of constraint based scheduling systems using artificial intelligence approach. Artificial Intelligence in Engineering, 6(1991)2, pp. 58–73.

[12] Leconte, V. Lelong: Constraint Programming – Génération Charme, Tutorial 6, 12th Int. Conference on Artificial Intelligence, June 1–6, 1992, EC2, Nanterre, 1992.

[13] Paul Harmon: Avignon '93, ISS Volume IX, No. 7, pp. 7–8, July 1993.

La programmation par contraintes

Une nouvelle méthode pour les problèmes complexes de planification et d'optimisation

La programmation par contraintes est une avancée en informatique permettant la réalisation rapide et à moindre coût d'applications conviviales pour des tâches complexes comme, par exemple, l'optimisation des charges de réseaux, la gestion d'emplois du temps, la gestion de production ou la génération de plans de route. Tout en expliquant ses concepts, l'article présente les avantages de la programmation par contraintes par rapport à d'autres approches, quelques systèmes opérationnels typiques et son potentiel en terme économique.