Convergence of GENET Guided Local Search for Constraint Satisfaction Problems

Q. Zhang, Z. Xu, E.P.K. Tsang, J.A. Ford, H. Li and J. Sun

Technical Report CSM-390
Department of Computer Science
University of Essex, Colchester, UK

September 2003

Convergence of GENET Guided Local Search for Constraint Satisfaction Problems

Q. Zhang, Z. Xu, E. P. K. Tsang, J. A. Ford, H. Li and J. Sun
Department of Computer Science, University of Essex, Colchester, CO4 3SQ, U.K.

Abstract

GENET and Guided Local Search (GLS) have been applied to a number of optimisation and search problems in the last decade. Despite their success, there has been little study of the properties of the GENET and GLS search mechanism. This paper provides a rigorous account of GENET local search for binary constraint satisfaction problems. This enables us to study the convergence of a slightly modified version of GENET. We have identified and established a condition under which the modified GENET algorithm will converge to the global optimum. This is a significant result as it enables us to identify problems which can be solved to optimality by the modified GENET method.

I. Introduction

Guided Local Search (GLS) is a meta-heuristic method for combinatorial optimisation [13][16][15][9]. Inspired by various ideas in Operations Research, GLS was generalised from GEneric NEural neTwork (GENET), which was proposed by Tsang and Wang in 1991 [1][2] and significantly extended by Davenport [3][4], for solving constraint satisfaction problems (CSPs) [5][6]. In GENET, a local search procedure is used to explore the search space. When the local search becomes trapped in a local optimum, the cost function is modified to help the local search procedure escape. GLS extends GENET by using problem-specific features to guide the search out of a local optimum.

GLS and its variants have been applied with great success to a number of combinatorial optimisation problems, including the radio link frequency assignment problem (RLFAP) [24][18], British Telecom's work force scheduling problem (WFS) [21][10], the travelling salesman problem (TSP) [11], function optimisation [22], processor configuration [17], the satisfiability (SAT) and MAX-SAT problems [19], rail traffic control [27], logic programming [28][29] and vehicle routing [23][26]. GLS has been incorporated into ILOG Dispatcher, a commercial package for vehicle routing problems [23]. With success in these applications, GLS and its variants are algorithms that combinatorial optimisation researchers cannot afford to ignore.

Given their experimental success, the issue of theoretical analysis of GENET and GLS has received some attention, e.g. see [25], which cast GENET as a Lagrangian method. The present paper attempts to provide a robust foundation for GENET and GLS. With constraint satisfaction problems, global solutions can be recognised (when no constraint is violated) once found. But in general, meta-heuristic methods cannot prove unsatisfiability in constraint satisfaction problems, or recognise global optimal solutions in optimisation problems. In this paper, we make our first attempt at studying the search mechanism of GENET and GLS. We prove that GENET can escape from a local optimum under certain conditions. This preliminary result should pave the way for further study of the working mechanism of GLS.

II. Constraint Satisfaction Problems

Constraint satisfaction is a general problem that appears in many applications [5][6]. Research in constraint programming has been active during the last two decades. Commercial products such as ILOG [30][31] and ECLiPSe [32] have been used in many commercial applications, including scheduling work for British Airways, French and British Rail, the Port of Singapore, Renault car manufacturing, etc.

In this paper, we define a constraint satisfaction problem (CSP) as a triple (Z, D, C), where

- $Z = \{\xi_1, \xi_2, \cdots, \xi_n\}$ is a finite set of variables,
- $D = \{1, 2, \cdots, m\}$ is a finite set of values,
- C is a set of constraints. Each constraint restricts certain simultaneous value assignments to a subset of the variables.

For convenience in the analysis (mainly to reduce necessary complexity in indexing), we assume, without loss of generality, that all values are available to all variables here. A label, denoted by (ξ, v) here, is a variable-value pair which represents the assignment of value $v \in D$ to variable $\xi \in Z$. We use $A = \{(\xi_1, v_1), \cdots, (\xi_n, v_n)\}$ to denote a simultaneous assignment in which the set of variables $\xi_1, \cdots, \xi_n$ is assigned the values $v_1, \cdots, v_n$, respectively. Given a CSP, the task is to find a simultaneous assignment which satisfies all the constraints in C. A CSP is called a binary CSP if each of its constraints involves two variables. It has been proved that any CSP can be transformed into a binary CSP. Therefore, in this paper, we only consider binary CSPs.

A. The Cost Function of a CSP

In GENET for the binary CSP [1][2], a simultaneous assignment $A = \{(\xi_1, v_1), \cdots, (\xi_n, v_n)\}$ to the binary CSP is represented as a 0-1 matrix:

$$x = \begin{pmatrix} x_{11} & x_{12} & \ldots & x_{1n} \\ x_{21} & x_{22} & \ldots & x_{2n} \\ \ldots & \ldots & \ldots & \ldots \\ x_{m1} & x_{m2} & \ldots & x_{mn} \end{pmatrix},$$

where

$$x_{vi} = \begin{cases} 1 & \text{if } v_i = v, \\ 0 & \text{otherwise.} \end{cases}$$

Obviously, only one element in each column of x takes the value of 1. For any $i \in \{1, 2, \cdots m\}$, $x_{vi} = 1$ represents the assignment of the value v to the variable ξ_i. In this paper, the i-th column of x is denoted by x_{*i}, i.e., $x = (x_{*1}, \cdots, x_{*n})$.

The search space Ω is defined as:

$$\Omega = \left\{ x = (x_{vi})_{m \times n} : x_{vi} \in \{0, 1\} \text{ and } \sum_{v=1}^{m} x_{vi} = 1 \text{ for } i = 1, \ldots, n \right\}. \tag{1}$$

Clearly, Ω contains all possible simultaneous assignments to the CSP.

The cost (or "energy") of a simultaneous assignment $x \in \Omega$ for the CSP is defined as:

$$E(x) = \sum_{u=1}^{m} \sum_{v=1}^{m} \sum_{i=1}^{n} \sum_{j=1}^{n} \omega_{vi,uj} x_{vi} x_{uj}, \tag{2}$$

where $\omega_{vi,uj}$ is a non-negative integer coefficient (called the *connection weight)* between the labels (ξ_i, v) and (ξ_j, u). For convenience, we assume that $\omega_{vi,uj} = \omega_{uj,vi}$ for any i, j, u, v. In GENET, the values of all weights $\omega_{vi,uj}$ satisfy the following condition:

$$\omega_{vi,uj} \begin{cases} = 0 & \text{if } i = j, \\ = 0 & \text{if } (\xi_i, v) \text{ and } (\xi_j, u) \text{ are compatible in the constraint set } C, \\ > 0 & \text{otherwise.} \end{cases} \tag{3}$$

The initial value of $\omega_{vi,uj}$ is set to 1 if and only if (ξ_i, v) and (ξ_j, u) are incompatible with each other. These values are, however, subject to increments during the search. It is evident that $E(x) = 0$ if and only if x satisfies all the constraints. In many applications, the problem might be overconstrained to the extent that no simultaneous assignment satisfies all the constraints. In this case, the CSP is regarded as being equivalent to the following optimisation problem:

$$\min E(x), \quad x \in \Omega. \tag{4}$$

$y \in \Omega$ is called a global optimum of (4) if $E(y) \leq E(x)$ for all $x \in \Omega$. y is called a local optimum if $E(y) \leq E(x)$ for any $x \in \Omega$ differing from y in only one column.

III. GENET Local Search

Before introducing GENET local search, we need to introduce the following operators:

Definition 1: [7][8] The WTA (winner-take-all) *operator* ϕ is a map: $\mathbf{R^m} \to \{\mathbf{0}, \mathbf{1}\}^{\mathbf{m}}$ in which, for any $a = (a_1, a_2, \ldots, a_m)^T \in \mathbf{R^m}$, $\phi(a) = (b_1, \cdots, b_m)^T$ is determined by

$$b_i = \begin{cases} 1 & \text{if } a_i = \min_{1 \leq k \leq m}\{a_k\} \text{ and } a_i < a_j \text{ for } j = 1, \cdots, i-1, \\ 0 & \text{otherwise.} \end{cases}$$

In other words, $b_i = 1$ iff a_i is the minimum element with lowest index in $\{a_1, \cdots, a_m\}$.

Definition 2: For each $i \in \{1, 2, \cdots, n\}$, the *move operator* Ψ_i on the variable ξ_i works as follows:

Move Operator: $y = \Psi_i(x)$

Input: $x \in \Omega$
Output: $y \in \Omega$

Step 1 For any $j \in \{1, 2, \cdots n\} \backslash \{i\}$, set $y_{*j} = x_{*j}$.

Step 2 Set

$$c_v = \sum_{u=1}^{m} \sum_{j=1}^{n} \omega_{vi,uj} x_{uj}, \quad v \in \{1, 2, \cdots m\}, \tag{5}$$

and let $c = (c_1, \cdots, c_m)^T$.

Step 3 Let v_i be the integer in $\{1, 2, \cdots, m\}$ such that $x_{v_i i} = 1$. If $c_{v_i} = \min_{1 \le k \le m} \{c_k\}$, then set $y_{*i} = x_{*i}$. Otherwise, $y_{*i} = \phi(c)$.

Obviously, $x = \Psi_i(x)$ if and only if $c_{v_i} = \min_{1 \le k \le m} \{c_k\}$.

The GENET local search works as follows:

$y =$ **GENET_LS(E, x)**

Input: Cost function $E(.)$ and a starting assignment $x = (x_{ij}) \in \Omega$
Output: an assignment $y = (y_{ij}) \in \Omega$

Step 1 $y^0 := x$, $t := 0$

Step 2 Set $NT = 0$,
For $i = 1, \cdots, n$ do
{
$z := \Psi_i(y^t)$;
if $z \neq y^t$, set $y^{t+1} := z$, $NT := NT + 1$ and $t := t + 1$;
}

Step 3 If $NT = 0$, return $y := y^t$ and stop. Else go to Step 2.

Evidently, $y^t \neq y^{t+1}$ in GENET_LS.

Theorem 1: If $\omega_{vi,ui} = 0$ for any $i \in \{1, 2, \cdots, n\}$ and for any $u, v \in \{1, 2, \cdots, m\}$, then GENET_LS will terminate in a finite number of steps and find a local optimum.

Proof: Consider two consecutive solutions y^t and y^{t+1} in the sequence generated by GENET local search. We first prove that $E(y^t) > E(y^{t+1})$. Note that y^{t+1} and y^t differ only in one column - say, the l^{th} column. We denote the non-zero elements of y^{t+1} and y^t in the l^{th} column by y^{t+1}_{dl} and y^t_{el}, respectively, i.e. $y^{t+1}_{dl} = y^t_{el} = 1$ and $d \neq e$. Then

$$\begin{aligned}
E(y^t) - E(y^{t+1}) &= \sum_{u=1}^{m} \sum_{v=1}^{m} \sum_{i=1}^{n} \sum_{j=1}^{n} \omega_{vi,uj} y^t_{vi} y^t_{uj} - \sum_{u=1}^{m} \sum_{v=1}^{m} \sum_{i=1}^{n} \sum_{j=1}^{n} \omega_{vi,uj} y^{t+1}_{vi} y^{t+1}_{uj} \\
&= \underbrace{[\sum_{u=1}^{m} \sum_{v=1}^{m} \sum_{i=1, i \neq l}^{n} \sum_{j=1, j \neq l}^{n} \omega_{vi,uj} y^t_{vi} y^t_{uj} - \sum_{u=1}^{m} \sum_{v=1}^{m} \sum_{i=1, i \neq l}^{n} \sum_{j=1, j \neq l}^{n} \omega_{vi,uj} y^{t+1}_{vi} y^{t+1}_{uj}]}_{=0} + \\
&\quad [\sum_{u=1}^{m} \sum_{v=1}^{m} \sum_{j=1}^{n} \omega_{vl,uj} y^t_{vl} y^t_{uj} - \sum_{u=1}^{m} \sum_{v=1}^{m} \sum_{j=1}^{n} \omega_{vl,uj} y^{t+1}_{vl} y^{t+1}_{uj}] + \\
&\quad [\sum_{u=1}^{m} \sum_{v=1}^{m} \sum_{i=1}^{n} \omega_{vi,uj} y^t_{vi} y^t_{ul} - \sum_{u=1}^{m} \sum_{v=1}^{m} \sum_{i=1}^{n} \omega_{vi,uj} y^{t+1}_{vi} y^{t+1}_{ul}] - \\
&\quad \underbrace{[\sum_{u=1}^{m} \sum_{v=1}^{m} \omega_{vl,ul} y^t_{il} y^t_{ul} - \sum_{u=1}^{m} \sum_{v=1}^{m} \omega_{vl,ul} y^{t+1}_{vl} y^{t+1}_{ul}]}_{=0 \text{ (Since } \omega_{vl,ul}=0)}
\end{aligned}$$

$$= 2(\sum_{u=1}^{m}\sum_{j=1}^{n}\omega_{el,uj}y_{uj}^{t} - \sum_{u=1}^{m}\sum_{j=1}^{n}\omega_{dl,uj}y_{uj}^{t}).$$

Since $\sum_{u=1}^{m}\sum_{j=1}^{n}\omega_{el,uj}y_{uj}^{t}$ and $\sum_{u=1}^{m}\sum_{j=1}^{n}\omega_{dl,uj}y_{uj}^{t}$ are c_e and c_d, respectively (by (5)), we have

$$\sum_{u=1}^{m}\sum_{j=1}^{n}\omega_{el,uj}y_{uj}^{t} - \sum_{u=1}^{m}\sum_{j=1}^{n}\omega_{dl,uj}y_{uj}^{t+1} > 0.$$

Thus

$$E(y^t) > E(y^{t+1}),$$

which means that the sequence of values $\{E(y^t)\}$ is strictly decreasing. Since the search space is finite, GENET_LS must therefore terminate in a finite number of steps.

Let y be the solution returned by GENET_LS. Then, for any $i \in \{1, 2, \cdots, n\}$, $y = \Psi_i(y)$. Thus when $z \in \Omega$ differs from y in one column, we have

$$\sum_{u=1}^{m}\sum_{j=1}^{n}\omega_{el,uj}z_{uj} - \sum_{u=1}^{m}\sum_{j=1}^{n}\omega_{dl,uj}y_{uj} \geq 0,$$

which means

$$E(z) \geq E(y).$$

Therefore y is a local minimum of $E(x)$. ■

IV. GENET Guided Local Search

Suppose that x^* is a local minimum obtained from the GENET local search procedure defined above. Suppose also that new connection weights are defined as follows:

$$\hat{\omega}_{ui,vj} = \begin{cases} \omega_{ui,vj} + \alpha x_{ui}^* x_{vj}^* & \text{if } i \neq j \\ \omega_{ui,vj} & \text{otherwise,} \end{cases} \tag{6}$$

where α is the *learning rate* (a positive integer that users need to set). The augmented function is then defined as

$$F(x) = \sum_{u=1}^{m}\sum_{v=1}^{m}\sum_{i=1}^{n}\sum_{j=1}^{n}\hat{\omega}_{ui,vj}x_{ui}x_{vj}. \tag{7}$$

Then we can apply the GENET local search procedure to $F(x)$, starting at x^*:

$$x^{**} = \mathbf{GENET\_LS(F, x^*)}. \tag{8}$$

By repeating the above procedure, we will generate a sequence of solutions and hope to obtain a global optimum.

V. Convergence of GENET

In general, metaheuristic methods, of which GLS is one, do not guarantee to find global optimum solutions. GLS is able to escape from local optima by means of penalty increments. However, there has been no guarantee until now that, after escaping from one local optimum, the method will find a better one. The following theorem shows that GENET can find better local minima under certain conditions.

Theorem 2: Suppose $\{y^0, y^1, \cdots, y^T\}$ is the sequence generated by the GENET_LS described in the above section, i.e., $x^{**} = y^T$ in (8). If $T > \alpha n(n-1)/2$, then $E(x^{**}) < E(x^*)$.

Proof: From (2), (7) and (6), we can show that

$$|E(x) - F(x)| = \left|\alpha \sum_{i \neq j}\sum_{u=1}^{m}\sum_{v=1}^{m} x_{ui}^* x_{vj}^* x_{ui} x_{vj}\right| \leq \frac{n(n-1)}{2}\alpha \tag{9}$$

for any $x \in \Omega$.

Since $\omega_{ui,vj}$ and $\hat{\omega}_{ui,vj}$ are integers, we have

$$F(y^i) - F(y^{i+1}) \geq 1$$

for $i = 0, 1, \cdots, T$. Then

$$F(x^*) - F(x^{**}) = \sum_{i=0}^{T-1} [F(y^i) - F(y^{i+1})] \geq T. \tag{10}$$

From (6),

$$E(x) < F(x) \tag{11}$$

for any $x \in \Omega$. Then

$$\begin{aligned} E(x^*) - E(x^{**}) &= F(x^*) - F(x^{**}) + [E(x^*) - F(x^*)] - [E(x^{**}) - F(x^{**})] \\ &\geq T - [E(x^{**}) - F(x^{**})], \text{using (10) and (11)}, \\ &\geq T - \frac{n(n-1)}{2}\alpha, \text{using (9)}. \end{aligned}$$

It follows that

$$E(x^*) - E(x^{**}) > 0.$$

■

The theorem above implies that, after the penalties are increased, GLS will be able to find a better local optimum if the number of GENET local search steps exceeds $\alpha n(n-1)/2$. When this condition is satisfied at all local optima, the modified GENET algorithm will be able to converge to the global optimum. For obvious reasons, this condition will never be satisfied at the global optimum.

The impact of this result is yet to be explored. In practice, the value of T is not under our control. However, it is measurable; as a matter of fact, it is monitored in [15], among other performance indicators. It should also be noted that the condition $T > \alpha n(n-1)/2$ is derived from the 'worst case', i.e. that in which the difference between $F(y^i)$ and $F(y^{i+1})$ assumes its lowest possible value of unity at every stage. Any differences greater than unity will cause a corresponding relaxation in the minimum value required for T. Furthermore, the learning rate parameter α is under our full control. It is plausible that α could be set, or adjusted dynamically, with the aim of satisfying the condition $T > \alpha n(n-1)/2$.

VI. Conclusion

This paper presents a mathematical formulation of GENET local search for binary constraint satisfaction problems. This enables us to study the convergence of GENET and GLS. We have proved that the local search in GENET can find a local optimum of the objective function. We have slightly modified the GLS in GENET to allow us to study its properties mathematically. We have proved that this modified version of GLS can move to a 'better' local optimum if the number of steps in the GLS phase is sufficiently large.

The formulation and slight modification of GLS and the convergence analysis provided in this paper pave the way for a further study of the mechanism and range of application of GLS. The number of moves between local optima (i.e. the values of T) may be an indicator of the hardness of the problem for the modified GLS. The condition above will potentially enable us to identify problems that can be solved to optimality by GLS. It may also help us to tune, or dynamically adjust, the learning rate parameter α, the only major parameter to be tuned in the modified GLS. The above result may also help us to guide the search. For example, looking ahead will potentially help us to pick neighbours that lead to a larger value of T, thus increasing our chances of improving on the previous local optimum. Besides, when the above condition is satisfied, x^* can be seen as in the same basin of attraction as x^{**}. The more local optima that satisfy this condition, the larger the basins of attraction will be. We would like to study the landscape of GLS in certain problems [33]. These are some of the issues that we would like to examine further.

References

[1] Wang, C.J. and Tsang, E.P.K., Solving constraint satisfaction problems using neural-networks, Proceedings, IEE Second International Conference on Artificial Neural Networks, 1991, 295-299

[2] Tsang, E.P.K. and Wang, C.J., A generic neural network approach for constraint satisfaction problems, in Taylor, J.G. (ed.), Neural network applications, Springer-Verlag, 1992, 12-22

[3] Davenport, A., Extensions and evaluation of GENET in constraint satisfaction, PhD Thesis, Department of Computer Science, University of Essex, Colchester, UK, 1997

[4] Davenport, A., Tsang, E.P.K., Wang, C.J. and Zhu, K., GENET: a connectionist architecture for solving constraint satisfaction problems by iterative improvement, Proc., 12th National Conference for Artificial Intelligence (AAAI), 1994, 325-330

[5] Tsang, E.P.K., Foundations of constraint satisfaction, Academic Press, London, 1993.

[6] Dechter, R., Constraint processing, Morgan Kaufmann, 2003

[7] Xu, Z.B., Jin, H.D., Leung, K.S., Leung, Y. and Wong, C.K., An automata network for performing combinatorial optimization, Nerocomputing, 47(2002), pp. 59-83.

[8] Xu, Z.B., Hu, G.Q., and Kwong, C.P., Asymmetric Hopfield-type networks: theory and applications, Neural Networks, 9(1996), pp. 483-501.

[9] Tsang, E.P.K., Wang, C.J., Davenport, A., Voudouris, C. and Lau, T.L., A family of stochastic methods for constraint satisfaction and optimization, The first International Conference on The Practical Application of Constraint Technologies and Logic Programming, London, April, 1999, pp. 359-383.
[10] Tsang, E.P.K. and Voudouris, C., Fast local search and guided local search and their application to British Telecom's workforce scheduling problem, Operations Research Letters, Elsevier Science Publishers, Amsterdam, Vol.20, No.3, March 1997, 119-127
[11] Voudouris, C. and Tsang, E.P.K., Guided local search and its application to the traveling salesman problem, European Journal of Operational Research, 113(1999), pp. 469-499.
[12] Voudouris, C. and Tsang, E.P.K., Guided local search, Chapter 7, in Glover, F. (ed.), Handbook of metaheuristics, Kluwer, 2003, 185-218
[13] Voudouris, C., Guided Local Search for Combinatorial Optimisation Problems, PhD Thesis, Department of Computer Science, University of Essex, Colchester, UK, May 1997
[14] Kilby, P., Prosser, P., and Shaw, P., A comparison of traditional and constraint-based heuristic methods on vehicle routing problems with side constraints, Constriaints, Kluwer Academic Publishers, Vol.5, No.4, 2000, 389-414
[15] Mills, P., Extensions to guided local search, PhD Thesis, Department of Computer Science, University of Essex, 2002
[16] Lau, T.L., Guilded Genetic Algorithm, PhD Thesis, Department of Computer Science, University of Essex, 1999
[17] Lau, T.L. and Tsang, E.P.K., Solving the processor configuration problem with a mutation-based genetic algorithm, International Journal on Artificial Intelligence Tools (IJAIT), (http://www.wspc.com.sg/journals/journals.html), World Scientific, Vol.6, No.4, December 1997, 567-585
[18] Lau, T.L. and Tsang, E.P.K., Guided genetic algorithm and its application to radio link frequency assignment problems, Constraints, Vol.6, No.4, 2001, 373-398
[19] Mills, P. and Tsang, E.P.K., Guided local search for solving SAT and weighted MAX-SAT problems, Journal of Automated Reasoning, Special Issue on Satisfiability Problems, Kluwer, Vol.24, 2000, 205-223
[20] Mills, P., Tsang, E.P.K. and Ford, J., Applying an extended guided local search to the quadratic assignment problem, Annals of Operations Research, Kluwer Academic Publishers, Vol.118, 2003, 121-135
[21] Azarmi N. and Abdul-Hameed W., Workforce scheduling with constraint logic programming, British Telecom Technology Journal, Vol.13, No.1, January, 1995, 81-94
[22] Voudouris, C., Guided Local Search – An illustrative example in function optimisation, BT Technology Journal, Vol.16, No.3, July 1998, 46-50
[23] Backer, B.D., Furnon, V., Kilby, P., Prosser, P. and Shaw, P., Solving vehicle routing problems using constraint programming and metaheuristics, Technical Report, GreenTrip Project, http://www.cs.strath.ac.uk/˜ps/GreenTrip/, 1997
[24] Bouju, A., Boyce, J.F., Dimitropoulos, C.H.D., vom Scheidt, G. and Taylor, J.G., Intelligent search for the radio link frequency assignment problem, Proceedings of the International Conference on Digital Signal Processing, Cyprus 1995
[25] Choi, K.M.F., Lee, J.H.M. and Stuckey, P.J., A Lagrangian Resconstruction of GENET, Artificial Intelligence, Vol.123, No.1-2, 2000, 1-39
[26] Kilby, P., Prosser, P., and Shaw, P., Guided local search for the vehicle routing problem with time windows, in Voss, S., Martello, S., Osman, I.H., and Roucairol, C. (eds.), Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization, Kluwer Academic Publishers, 1999, 473-486
[27] Jose, R. and Boyce, J., Appication of connectionist local search to line management rail traffic control, Proceedings, Practical Applicaiton of Constraint Technology (PACT'97), London, April 1997
[28] Lee, J.H.M. and Tam, V.W.L., A framework for integrating artificial neural networks and logic programming, International Journal on Artificial Intelligence Tools, Vol.4, Nos.1and2, June 1995, 3-32
[29] Stuckey, P. and Tam, V., Semantics for using stochastic constraint solvers in constraint logic programming, Journal of Functional and Logic Programming, January 1998
[30] http://www.ilog.fr
[31] Puget, J-F., Applications of constraint programming, in Montanari, U. and Rossi, F. (ed.), Proceedings, Principles and Practice of Constraint Programming (CP'95), Lecture Notes in Computer Science, Springer Verlag, Berlin, Heidelberg and New York, 1995, 647-650
[32] Lever, J., Wallace, M. and Richards, B., Constraint logic programming for scheduling and planning, British Telecom Technology Journal, Vol.13, No.1., Martlesham Heath, Ipswich, UK, 1995, 73-80
[33] Reeves, C.R., Landscapes, operators and heuristic search, Annals of Operations Research, Vol.86, 1999, 473-490