Research Journal of Applied Sciences, Engineering and Technology 5(5): 1736-1739, 2013
DOI:10.19026/rjaset.5.4930
ISSN: 2040-7459; e-ISSN: 2040-7467
© 2013 Maxwell Scientific Publication Corp.
Submitted: July 27, 2012 Accepted: September 03, 2012 Published: February 11, 2013

Research Article

Research on Mechanism of Variable Event in CLP System

Jing Wan, Likang Tian, Junkai Yi and Yubin Zhou
Department of Information Science and Technology, Beijing University of
Chemical Technology, Beijing 100029, China

Abstract: Constraint inference engine as the core part of constraint logic programming system comprises variable set, temporary container and constraint filter and inference engine and adopts branching strategy, exploration strategy and node backtracking strategy to complete the inference task. This study introduces variable event to reduce the triggering times of constraint filter and analyzes the trigger probability of inference engine. The experiments show that variable event setting can enhance the efficiency of constraint inference engine.

Keywords: Constraint logic programming, inference engine, variable event

INTRODUCTION

Constraint logic programming, which combines the advantages of consistency algorithms and heuristic search algorithms in artificial intelligence, is one of the important paradigms for solving discrete domain constraint problems (Hendler *et al.*, 2006).

Generally, the basic strategy adopted by constraint logic programming is binary tree search, which begins from root node in a prescribed way. Once a leaf is found, an answer is found. The establishment of binary tree is a modeling process of practical problem, which involves a series of logic variables as the mathematical foundation for problem description. Each variable has its own scope referred to as finite field. Constraint conditions that should be met by problem description can be expressed as inter-variable logic constraint, which is a constraint filter in the search process. What needs to be solved next is how to get a smaller finite field so as to satisfy constraint filter. And then the inference engine of constraint logic programming is initiated. Inference engine can perform binary tree search according to the prescribed strategy in a temporary container. Interactions of variable set, constraint filter, temporary container and inference engine collectively constitute constraint inference engine (Miguel *et al.*, 2008).

Efficiency improvement of constraint inference engine has become one of the main research directions for constraint logic programming (Roman *et al.*, 2010). In order to improve the search efficiency of constraint logic programming system, enhancement of the pruning ability of constraint inference engine is the main approach (Alan *et al.*, 2009), which involves such strategies as branch algorithm (Asma *et al.*, 2011), exploration orientation (Samir *et al.*, 2010) and node backtracking algorithm (Mesyagutov *et al.*, 2012). Based on these strategies, this study proposes the classification of variable compression event, which reduces the accelerated cycles of trigger times of constraint filter in the inference engine and improves search efficiency.

VARIABLE EVENT

Search efficiency analysis of constraint inference engine: Constraint inference engine is composed of variable set, constraint filer, temporary container and inference engine. Constraint inference engine has branching strategy, exploration strategy and node backtracking strategy, which prescribes the behaviors of binary tree search. In constraint logic programming, constraint inference engine is the core, whose efficiency improvement is important to enhance the search efficiency. This study proposes the classification of variable events in the purpose of improving search efficiency by reducing the search time.

Each inference of constraint inference engine does not require the involvement of all constraint filters. Only when a variable's finite field is changed will the constraint filter related to this variable be triggered and participate in the inference. This is the trigger mechanism of filter for all constraint logic programming systems. A reduction in finite field of a variable is defined as a variable event, denoted as Ex. In order to further enhance the search efficiency, variable event needs to be processed, by classifying the constraint filters with respect to a variable according to the type of variable event. Only when a specific variable event occurs will the related constraint filter be

Corresponding Author: Junkai Yi, Department of Information Science and Technology, Beijing University of Chemical Technology, Beijing 100029, China

triggered. Variable event setting does not increase memory consumption but avoids many unnecessary trigger of constraint filter.

Type of variable event: Suppose the initial discrete range of x is $\{w_1, w_2,..., w_n\}$, then the range of x will be reduced after w_k is filtered out by constraint inference engine. Variable x before and after the filtering of constraint inference engine can be expressed by Formula (1) below:

$$x \in \{w_1, w_2,\ldots, w_n\} \rightarrow x \in \{w_1,\ldots w_{k-1}, w_{k+1},\ldots, w_n\} \quad (1)$$

As we can see from Formula (1), when some values are arbitrarily filtered, the form of the finite field of a variable is changed. Then event type E_x of variable x is the variation of finite field; E_x is defined as DOMCHG by definition 1.

Definition 1: For x and its finite field D_x, if $D_x' \subset D_x$ and $D_x' \neq \Phi$, then E_x is defined as DOMCHG in the process of $D_x \rightarrow D_x'$.

As shown by Formula (2) and (3), if value w_1 or w_n is filtered from variable x, constraint inference engine will define the type of variable event E_x as BOUND, with a changed boundary of variable range:

$$x \in \{w_1, w_2,\ldots, w_n\} \rightarrow x \in \{w_2, \ldots, w_n\} \quad (2)$$

$$x \in \{w_1, w_2, \ldots, w_n\} \rightarrow x \in \{w_1, \ldots, w_{n-1}\} \quad (3)$$

For finite field D, suppose max (D) and min (D) are respectively the maximum value and the minimum value and then there is definition 2.

Definition 2: For x and its finite field D_x, if $D_x' \subset D_x$, $D_x' \neq \Phi$ and either of the following two conditions are satisfied: max (D_x') <max (D_x) or min (D_x') >min (D_x), then E_x is defined as BOUND in the process of $D_x \rightarrow D_x'$.

If the range of variable x is reduced to only one value from many discrete values, constraint inference engine will define the type of variable event E_x as SINGLE, which is expressed by Formula (4):

$$x \in \{w_1, w_2, \ldots, w_n\} \rightarrow x \in \{w_k\} \ (1 \leq k \leq n) \quad (4)$$

For finite field D, suppose Size (D) the represents number of possible values in D, then there is definition 3.

Definition 3: For x and its finite field D_x, if $D_x' \subset D_x$, Size (D_x') = 1, then E_x is defined as SINGLE in the process of $D_x \rightarrow D_x'$.

Therefore, variable event E has three types: SINGLE, BOUND and DOMCHG. From the above three definitions, the following two inferences can be obtained:

Inference 1: For x and its finite field D_x, if $D_x' \subset D_x$, $D_x' \neq \Phi$ and E_x is defined as SINGLE in the process of $D_x \rightarrow D_x'$, then E_x can also be defined as DOMCHG and BOUND simultaneously.

Inference 2: For x and its finite field D_x, if $D_x' \subset D_x$, $D_x' \neq \Phi$ and E_x is defined as BOUND in the process of $D_x \rightarrow D_x'$, then E_x can be defined as DOMCHG.

Here we can see that E_x can be defined as more than one variable event simultaneously, but that is not to say that E_x is changeable. What is true is that a variable defined as a certain event type can contain other events simultaneously. For variable x, SINGLE $\subseteq$ BOUND $\subseteq$ DOMCHG, SINGLE is contained in BOUND and DOMCHG, while BOUND is contained in DOMCHG.

VARIABLE EVENT SETTING

When constraint filters are incorporated into constraint inference engine, relevant variables need to be standardized and initialized so that the variables can meet the requirements of constraint inference engine. In the initialization step, variable event setting can be performed according to actual contents of constraint algorithm.

From the above analysis we have known that variable event can have three types: SINGLE, BOUND and DOMCHG. In this section, variable event setting for a specific constraint filter is discussed by taking linear filter as an example. Formula (5) represents a linear constraint filter; ai is a non-zero integer constant:

$$\sum_{i=0}^{i<n} a_i x_i = A \quad (5)$$

where, A is an integer constant. The left side of Formula (5) can be divided into two parts according to the sign of a_i, as shown in Formula (6):

$$A = \sum_{i=0}^{i<n} a_i x_i = \sum_{a_i>0}^{i<n} a_i x_i + \sum_{a_j<0}^{j<n} a_j x_j \quad (6)$$

Suppose max(x) and min(x) respectively represents the maximum value and minimum value of x. Then from Formula (6), we can see that the ran ge of x_i is:

$$\frac{A - \sum_{a_i>0, i\neq k}^{i<n} a_i \max(x_i) - \sum_{a_j<0, j\neq k}^{j<n} a_j \min(x_j)}{a_k} \le x_k \le \frac{A - \sum_{a_i>0, i\neq k}^{i<n} a_i \min(x_i) - \sum_{a_j<0, j\neq k}^{j<n} a_j \max(x_j)}{a_k}$$

$(a_k>0,\ k<n)$ (7)

$$\frac{A - \sum_{a_i>0, i\neq k}^{i<n} a_i \min(x_i) - \sum_{a_j<0, j\neq k}^{j<n} a_j \max(x_j)}{a_k} \le x_k \le \frac{A - \sum_{a_i>0, i\neq k}^{i<n} a_i \max(x_i) - \sum_{a_j<0, j\neq k}^{j<n} a_j \min(x_j)}{a_k}$$

$(a_k<0, k<n)$ (8)

If n - 1 variables have definite single value and only 1 variable has changeable value range, then constraint expressed in Formula (9) will come into effect:

$$x_i = (A - \sum_{j\neq i}^{j<n} a_j x_j) / a_i \ (i<n) \tag{9}$$

Constraint filter expressed in Formula (9) requires Formula (6)-(8) to reduce the range of x_i. Only when the boundary of x_i is changed will these constraints come into effect. Therefore, x_i variable event can be defined as BOUND, by Formula (10):

$$\mathrm{E}(x_i) = \mathrm{BOUND} \tag{10}$$

Variable event setting for constraint filter must be determined by the algorithm of this filter. For the same filter, variable event varies due to variable algorithms. A good filter algorithm has excellent pruning ability to reduce the search space and it can substantially enhance the search efficiency in combination with variable event.

Trigger times analysis of variable event: The trigger probability of variable event is mainly reflected in the trigger times of constraint filter. Suppose the finite field of variable x is $D_x = \{w_1, w_2,\ldots, w_k\}$, where $w_1<w_2<\ldots<w_k$ and they are all integer numbers. Then, when DOMCHG happens to x, possible compression of D_x will occur as follows:

$$C_k^1 + C_k^2 + \ldots + C_k^{k-1} = 2^k - 2 \tag{11}$$

At this moment, the probability of x triggering related filter is O (2^k). When BOUND happens to x, possible compression of D_x will occur as follows:

$$k + (k - 1) + \ldots + 1 - 1 = \frac{k(k+1)}{2} - 1 \tag{12}$$

At this moment, the probability of x triggering related filter is (k^2). When SINGLE happens to x, there are k types of possible compression of D_x; O (k) is the probability.

If DOMCHG is a total probability event, which can be defined as:

$$P\,(\mathrm{DOMCHG}) = 1 \tag{13}$$

Then occurrence probability of SINGLE and BOUND is:

$$P(\mathrm{BOUND} \mid \mathrm{DOMCHG}) = \frac{\frac{k(k+1)}{2} - 1}{2^k\text{-}2} = \frac{k^2 + k - 2}{2^{k+1}\text{-}4} \tag{14}$$

$$P(\mathrm{SINGLE} \mid \mathrm{DOMCHG}) = \frac{k}{2^k\text{-}2} \tag{15}$$

As we can see from the above formula, for a certain variable, the greater the value of k, the less the occurrence of SINGLE than that of DOMCHG and BOUND and the less the occurrence of BOUND is than that of DOMCHG.

Table 1: Comparison of trigger times before and after the introduction of variable event

Sample	Nqueen (100)	Magic square (5)	Hamilton	Photo	Robin	Larry	Bridge
Trigger times before the introduction	102	87	233	302	156	126	119
Trigger times after the introduction	98	76	201	237	69	112	67

Table 2: Comparison of operation time before and after the introduction of variable event (operation time: s)

Sample	Nqueen (100)	Magic square (5)	Hamilton	Photo	Robin	Larry	Bridge
T before the introduction	0.53	0.15	22.13	26.05	12.47	3.98	1.14
T after the introduction	0.45	0.14	15.68	18.46	5.750	2.54	0.67

According to the previous method for constraint filter trigger, the probability of *x* triggering related SINGLE or BOUND filter is O (*2k*). The probability analysis of variable event occurrence shows that: when DOMCHG happens to *x*, BOUND and SINGLE constraint filter will not be triggered; when BOUND happens to *x*, SINGLE constraint filter will not be triggered.

EXPERIMENTAL RESULT ANALYSIS

After the introduction of variable event, the constraint filters are much less triggered during the search process. The experimental environment for constraint logic programming is as follows: HP2210b, T7300, duo CPU 2.0 G, 1 G Memory.

For the sake of comparison, classical problems are used for constraint solving. The experimental results are shown in Table 1 and 2. Operation time in the experiment includes the display time of screen and therefore the actual solution time is probably less. The introduction of variable event is of great help for improving the search efficiency of constraint logic programming and reducing unnecessary trigger of constraint filter.

CONCLUSION

Constraint inference engine as the core part of constraint logic programming comprises variable set, temporary container and constraint filter and inference engine and adopts branching strategy, exploration strategy and node backtracking strategy to complete the inference task. Search time is one of the main performance indexes of constraint logic programming system. This study introduces variable event to reduce the triggering of constraint filter and analyzes the trigger probability of inference engine after the introduction of variable event. Variable event can be divided into three types: SINGLE, BOUND and DOMCHG. Accordingly, constraint filter can be also divided into three types depending on the type of variable event that triggers the constraint filter: SINGLE, BOUND and DOMCHG. We have experimentally verified that variable event setting can enhance the efficiency of constraint inference engine.

REFERENCES

Alan, M.F, H. Brahim, K. Zeynep, M. Ian and W. Toby, 2009. Filtering algorithms for the multiset ordering constraint. Artif. Intell., 73(2): 299-328.

Asma, L., L. Pierre and H. Mohamed, 2011. Climbing depth-Bounded adjacent discrepancy search for solving hybrid flow shop scheduling problems with multiprocessor tasks. 8th International Conference on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems, pp: 117-130.

Hendler, J., H. Kitano and B. Nebel, 2006. Handbook of Constraint Programming. Elsevier, Holland, Amsterdam.

Mesyagutov, M., G. Scheithauer and G. Belov, 2012. LP bounds in various constraint programming approaches for orthogonal packing. Comp. Oper. Res., 39(10): 2425-2438.

Miguel, A.S., G. Antonio and B. Roman, 2008. Introduction: Special issue on constraint satisfaction techniques for planning and scheduling problems. Eng. Appl. Artif. Intell., 21(5): 679-682.

Roman, B., A.S. Miguel and R. Francesca, 2010. Constraint satisfaction techniques in planning and scheduling. J. Intell. Manuf., 21(1): 5-15.

Samir, L., B. Patrice and L. Nicolas, 2010. Advanced generic neighborhood heuristics for VNS. Eng. Appl. Artif. Intell., 23(5): 736-764.