УДК 519.95(6)

Informatics and Mathematical Methods in Simulation
Vol. 7 (2017), No. 1-2, pp. 135-138

SUBMISSION OF ALGORITHM FOR WORKS SEQUENCES FINDING WITH PETRI NET

A.N. Tynynyka

Odesa National Polytechnic University,
1 Shevchenko Str., Odesa, 65044, Ukraine; e-mail: lorans53@mail.ru

An easy and close to the optimal heuristic algorithm to solve a problem of scheduling theory with some restriction is offered. Such algorithms it is useful to represent of a secure Petri net. The figure steps for solving the problem are present using graphic notation in classical Petri nets plus additional extensions. The reasons for the fall in popularity of Petri nets and the ways of development of its expressive possibilities analyzed. In particular, it will contribute to creating a more complete standard models library with a clear graphical representation.

Keywords: theory of schedules, the sequence of works, algorithm, Petri net

Introduction

Separate process describes by a program that can be written in many languages. Petri nets successfully represent the structure of the control programs, convenient for program. To calculate values they are not designed, but they are designed to simulate instructions and the flow of information.

The standard way of presenting control programs structure is a block diagram algorithms. Block diagram represents the flow of control in a program.

Each algorithm can be depicted as a flowchart and then programmed. Thus, describing the proposed algorithm in the article as a Petri net, we represent and Petri net and, indeed, the follow-up program.

But first, let's solve one application task.

Tusk formulation

State the task of scheduling theory with this formulation.

One performer must perform n works with known duration of $l_i, i = 1...n$. In the specified calendar dates inspection work carried out, and only completed work accepts. Fix the following sequence works to minimize performed but not accepted works (not performed as a whole at the time of acceptance). This problem makes sense if unfinished deadline work fined.

For the sake of simplicity (not narrowing its mission), assume all integer values. Works will be planning as a whole, without gaps in the performance of any work.

Main body

To solve the problem proposed heuristic (not strict) algorithm acceptable computational complexity.

Sequence of operations when solving:

1. Distribute all upcoming work in order of duration l_i.
2. First step: $k=1$. Time step (the moment) $t=0$. When this $U_1=W, V_1=\varnothing$, where W is the quantity of numbers all works; V_k – the quantity of numbers planned activities; U_k – the quantity of numbers have not yet scheduled works.
3. Define the quantity of candidates B_k (step k) among the unplanned works: $i \in B_k$; if $t+l_i < c_i, i \in U_k$; c_i – planned completion date.
4. If $B_k=\varnothing$, turn to p. 9.
5. Calculate the value of the reserve q_i and the amount of losses (fine) s_i: $q_i = c_i-(t+l_i)$ and $s_i=(t+l_i)-c_i$, i.e. if $c_i > (t+l_i)$, include the difference to the reserve, and if $c_i < (t+l_i)$ – add to the amount of losses.
6. Calculate the index value for each work candidate: $J_i = d \cdot s_i + (1-d)q_i$, $i \in B_k$, where $d \in [0,1]$ – weighting factor.
7. Choose the following work in according to minimum index: $v_k = \min\{J_i\}, i \in B_k$.
8. Eliminate the planned work of the unplanned for the next step: $U_k+1 = U_k \setminus \{v_k\}$. Increase the next step time on the duration of planed work: $t = t + l_{yk}$. Turn to step 11.
9. When there are no candidates, find an unplanned work with minimum time delay in the end (a fine proportional to the duration of delay).
10. Repeat pp. 3, 5, 6, 7, 8.
11. If the scheduled less than $n-1$ work, perform the following step: $k=k+1$, referring to p. 3. Otherwise, proceed to the next item.
12. At the final step plan to the rest of unplanned last work.

Submitted algorithm requires less on average than n^3 elementary action.

After obtaining sequences of operations typically to build a graphic block diagram of the algorithm.

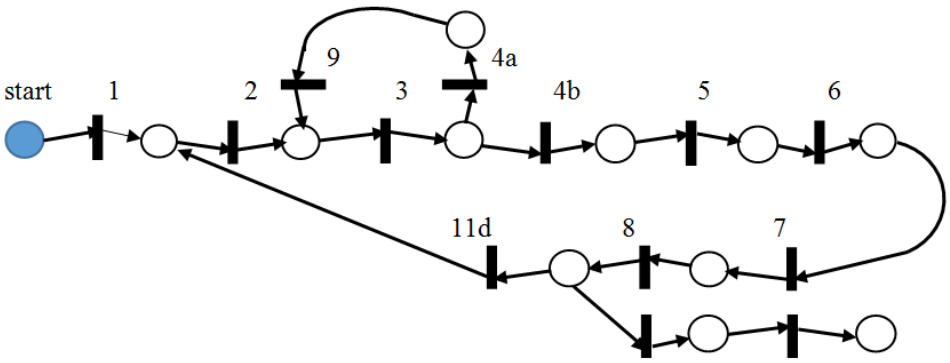

Fig. 1. Modeling with right secure Petri net of operations sequence when solving the task

Block diagram much like Petri nets [1]. Block diagram is visualized in the form of two types nodes (the decision and computation) and arcs between them (like Petri nets). However, in Petri nets transitions modeled actions, whereas in flowcharts nodes modeled actions. Thus, the correct translation of flowcharts in network nodes replaces on transitions network, and arc flowchart – on position of the network. Each arc flowchart corresponds exactly one position network. Flowchart nodes appear differently depending on the type: calculation or decision.

The fig. 1 shows these rules based Petri net that is equivalent to a possible flowchart for the proposed solution algorithm formulated task. Transitions numbering corresponds to the rows numbering of operations sequence when solving. Each position here has only one output

transition, except the one that preceded the adoption of the decision; such positions have two output transitions corresponding to the true and the false value of the predicate.

Transitions are associated with the actions of the program: computing and acceptances of decisions. To interpret the Petri net must interpret each passage. It should also be noted that transitions for calculations have one input and one output; the transition represents a calculation could not be in conflict, because his input position is not input for any other transition. The same action that is associated with the adoption of the decision, introduces a network conflict.

From the perspective of semantic interpretation in terms of simulated problem positions are associated with data, states or goals, while transitions are associated with operations (processes, procedures), that perform data conversion, implement response to input events, etc. Assumed that the transitions are performed instantly.

In Petri nets to represent the momentum used markers (tags, chips), which can be placed in position. Making a marker in the starting position breeds chain positives transitions. Move of marker from position to position reflects the dynamics of the network, and their initial location determines the specific process instance. The current distribution of markers in the positions sets the current state of the execution process modeled with Petri net.

One of the most important properties of Petri nets which simulates a real device or process is safety. The position of the Petri net is safe if the number of markers it never exceeds one. Petri net is safe if all net positions are safe.

The position is interpreted as a condition. The condition of being a logical expression is or true (represented with the token at position), or false (no token); multiple markers have no interpretation [2]. Thus, if network interpreted as conditions and events, marking each position must be safe.

Summation of the algorithm construction. The proposed algorithm is heuristic and does not guarantee an optimal solution, but studies with Monte-Carlo method indicate that decisions are close to optimal.

The average efficiency of the algorithm depends on the weighting factor d. Analysis made it possible to give a general recommendation: at low tension works expedient $d \approx 0.75$, at high – $d \approx 0.4$. Low tension corresponds to the more lengthy execution of works (when someone has time any delay to catch up later).

Conclusion

Historically, the Petri net was one of the first formal models intended for processes model specification. The popularity of the Petri net model, which was very high before the 1990's, is now dropped. The main reason for this is, apparently, in the weakness of submission means operational semantics. And the model does not disclose the questions relating to the execution environment. In view of the insufficiency of expressive possibilities Petri net and its available extensions, for example, chromatic, inhibitory, hierarchical Petri nets, etc., should be thinking about further expansion of models and use it as the basis for other languages, such as language YAWL (Yet Another Workflow) for modeling business processes first of all [3, 4, 5] or in language Workflow Net, where for some of the transitions require explicit writing expressions to complement the structural model of Petri nets. This, in particular, is in need of unambiguous choice of transitions requiring control logic, relevant operations "or" and "exclusive or". Workflow Net language allows to express the formal semantics of workflows, i.e. clearly defines the rules of transition between nodes. But in all cases, when there are preconditions to the model of Petri nets refuse is not worth it.

Representation of the actions sequence in the fig. 1 uses graphic notation in classical Petri nets with small additional extensions, namely [5]:

1. It is present position in it, which does not have input arcs (initial position);
2. It is present position in it, which does not have output of arcs (end position).

The formal apparatus of Petri nets and their extensions [1] gives the possibility to create a library of generic models with a clear graphic representation. This last, in our view, the most important thing is – for the elements of Petri nets offered clear and easy-to-use graphic notation, satisfying aesthetic orientation of different users that largely determined the attraction of this model for developers, at least because of this model deserves no constriction and increased use. This article, in particular, performs such role.

References

1. Котов, В.Е. Сети Петри / В.Е. Котов. – М.: Наука, 1984. – 160 с.
2. Sheer, A. ARIS-Business Process Frame / A. Sheer. – Springer, 1999. – 170 p.
3. Aalst, W. Yet Another Workflow Language / W. Aalst, A. YAWL Hofstede // Information Systems. – 2005. – No.30(4). – PP. 245-275.
4. Aalst, W. The application of Petri Nets to Workflow management / W. Aalst // Journal of Circuits, System and Computers. – 1998. – No.8(1). – PP. 21-66.
5. Yet Another Workflow Language. http://www.Yawl-system.com.

ПОДАННЯ АЛГОРИТМУ ВИЗНАЧЕННЯ ПОСЛІДОВНОСТІ РОБІТ МЕРЕЖЕЮ ПЕТРІ

О. М. Тининика

Одеський національний політехнічний університет,
просп. Шевченка, 1, Одеса, 65044, Україна; e-mail: lorans53@mail.ru

Запропоновано недалекий від оптимального евристичний алгоритм для розв'язання однієї задачі теорії розкладу з обмеженням припустимої обчислювальної складності. Такі алгоритми зручно и подавати безпечною мережею Петрі. Звернута увага на доцільність використання традиційних і розширених мереж Петрі для зображення подібних структур. На рисунку послідовність дій при рішенні задачі зображена за допомогою графічної нотації класичної мережі Петрі плюс додаткові розширення. Обговорені причини падіння популярності мереж Петрі і шляхи розвитку її виразних можливостей. Зокрема, цьому буде сприяти створення більш повної бібліотеки типових моделей з наочним графічним представленням.
Ключові слова: теорія розкладу, послідовність робіт, алгоритм, мережа Петрі

ПРЕДСТАВЛЕНИЕ АЛГОРИТМА ОПРЕДЕЛЕНИЯ ПОСЛЕДОВАТЕЛЬНОСТИ РАБОТ СЕТЬЮ ПЕТРИ

А.Н. Тынынка

Одесский национальный политехнический университет,
просп. Шевченко, 1, Одесса, 65044, Украина; e-mail: lorans53@mail.ru

Предложен несложный и близкий к оптимальному эвристический алгоритм решения одной задачи теории расписаний с ограничением. Алгоритмы такого рода удобно представлять безопасной сетью Петри. Обращено внимание на целесообразность использования традиционных и расширенных сетей Петри для изображения подобных структур. На рисунке последовательность действий при решении задачи представлена с использованием графической нотации классической сети Петри плюс дополнительные расширения. Обсуждены причины падения популярности сетей Петри и пути развития её выразительных возможностей. В частности, этому будет способствовать создание более полной библиотеки типовых моделей с наглядным графическим представлением.
Ключевые слова: теория расписаний, последовательность работ, алгоритм, сеть Петри