Nataliya GULAK[1], Olesya YAKOVENKO[2]

Scientific supervisor: Nataliya GULAK

SYSTEM WSPOMAGANIA DECYZJI W WARUNKACH PRZECIĄŻENIA

Streszczenie: W artykule rozpatruje sie system podejmowania decyzji w warunkach dużego obciążenia. Zastosowano metodę wyznaczenia współczynników doniosłości oraz zmodyfikowany algorytm wyboru trasy – co pozwala skrócić czas dla podjęcia decyzji.

Słowa kluczowe: system wspomagania decyzji, przeciążenie, granica celna, algorytm routingu

DECISION SUPPORT SYSTEM IN CONDITIONS OF THE HEAVY LOAD

Summary: In this paper we present an optimal planning process of decision support system in heavy overload conditions of passengers, goods and vehicles movement through the customs.

Keywords: decision support system, overload, customs border, routing algorithm.

Introduction

In terms of a political situation that prevailed in Ukraine, numbers of security checkpoints has grown dramatically. Increasing monitoring efficiency over the movement of products and vehicles through the customs border is one of the elements of country's socio-economic interests protection. Reducing the time of vehicles service on security checkpoints is one of the tasks on the way to improve work in terms of passenger-product traffic amounts change. To solve the task of service time reducing we present you decision support system (DSS) model in conditions of the heavy overload [1,2].

Let's consider work planning of DSS in conditions of the overload. Since it is possible to distribute the solution of the task over the system nodes in the process of distributed DSS functioning, then the simultaneous appearance of the multiple tasks that require maitenance can not be ruled. Therefore it is necessary to establish the optimal, in terms of some criterion, schedule of their solution, that is focus on the task that has highest priority. In general, the task of creating schedules is the following.

[1] National Aviation University, email: gulak_n@ukr.net
[2] National Aviation University

Let $T = \{t_i\}$ –variety of the tasks that require maintenance, η_i –a function of fine for a task t_i staying in the system. It is necessary to find a task permutation Π, where Π(k) = i, if *ith* task needs to be solved by *kth* on account that the total function must be minimal, that is

$$\sum_{i=1}^{n} \eta_i \to \min . \qquad (1)$$

The task of creating schedules in the (1) formulation belongs to a *NP*-full class which is traditionally considered intractable. Given the fact that the task is often solved in real time, it is necessary to use such methods that have the lowest computational complexity. One of these methods is method [3], computational complexity which in the worst case ie equal to $O(n2)$. In this method the weighted average completion time of tasks is used as a schedule's performance indicator (criterion of evaluation).
To formalize the task Pof scheduling let's introduce the following notations:
$(F_j <)$ –task system, where F is indexed set of n tasks, $n \geq 0$, and < – partial order relation (the ratio of precedence) given by F;
$I(P)$ –*F*index set;
$T_j, j \in I(P)$ – element of *F*;
$\tau_j, j \in I(P)$ – a need for task T_j in time maintenance;
$\omega_j, j \in I(P)$ – the cost of task T_j staying in the system (the importance of task).
Values τ_j and ω_j are calculated either as an average values in the presence of statistical data or with appropriate accessory functions that are received in an expert way.
Since we are considering a single processor, the schedules will considered as tasks' indexes permutation.
Permutation $\alpha = \alpha_1, \alpha_2, ..., \alpha_n$ of the $I\ (P)$elements is compatible with P if $T_j < T_j^{'}$ pulls $k < k^{'}$,where $\alpha_k = j$ and $\alpha_k = j^{'}$.
Permutation $\alpha = \alpha_1, \alpha_2, ..., \alpha_n$ of the $I(P)$ elements determines the schedule in a natural way: task T_{α_1} is executed first, task T_{α_2} is executed second, and so on. The average weighted completion time for scheduling a task that is determined α permutation is calculated as:

$$\overline{T(\alpha)} = \sum_{j=1}^{n} \omega_{\alpha_j} \left(\sum_{i=1}^{j} \tau_{\alpha_i} \right). \qquad (2)$$

Permutation α (schedule) is optimal if α is compatible with P and at the same the minimum $\overline{T(\alpha)}$ is attained among all permutations that are compatible with P. Let $p_j = \frac{\omega_j}{\tau_j}$ for all j from $I\ (P)$.

Theorem 1.1 [4]
Let P - streamline task. Then the permutation $\alpha = \alpha_1, \alpha_2, ..., \alpha_n$ is optimal in relation to Pif and only if, when $p_{\alpha_1} \geq p_{\alpha_2} \geq ... \geq p_{\alpha_n}$.
This theorem essentially determines the content of permutation finding algorithm.
Let $U \subseteq I(P)$ and $U \neq \varnothing$. Determine $\tau(U)$, $\omega(U)$ and $p(U)$ in such way:

$$\tau(U) = \sum_{j \in U} \tau_j \ , \ \omega(U) = \sum_{j \in U} \omega_j \ , \ p(U) = \frac{\omega(U)}{\tau(U)} .$$

Definition 1.1.
Set $U \subseteq I(P)$ is called an initial set in relation to P if:
1) $U \neq \varnothing$;
2) if $T_j < T_j^{'}$ then $j \in U$ pulls $j^{'} \notin U$.
Let Y – class of initial sets (in relation to P). Determine the number $p^* = \max_{U \in Y} p(U)$
Definition 1.2.
Set $U \subseteq I(P)$ is called a p - maximal set in relation to P, if:
1) $U \in Y$;
2) $p(U) = p^*$;
3) if $V \in Y$; $p(V) = p^*$ and $V \subseteq U$ then $V = U$.
Let $U \subseteq I(P)$ and define F / U as $F / U = \{T_j \mid T_j \in F, j \in U\}$.
Denote by H_j the set, that consists of index j and indexes i of all tasks for which $T_i < T_j$. A construction of an optimal permutation in scheduling task P can be accomplished in the following manner. Each task T_i is assigned a number $p(H_j)$, and then index l is defined such that $p(H_l) \geq p(H_j)$ for $j \in I(P)$, and $p(H_l) \geq p(H_j)$ for all indexes are $j \neq l$. Since T_l is a precursor of all tasks, indexes of which are included in H_l , there is an optimal in relation to P permutation α such that $\alpha = \alpha / (H_l - \{l\}); \alpha / (I(P) - H_l)$. In such way the task P can be decomposed into two smaller tasks, such as: $P / (H_l - \{l\})$ and $P / (I(P) - H_l)$.

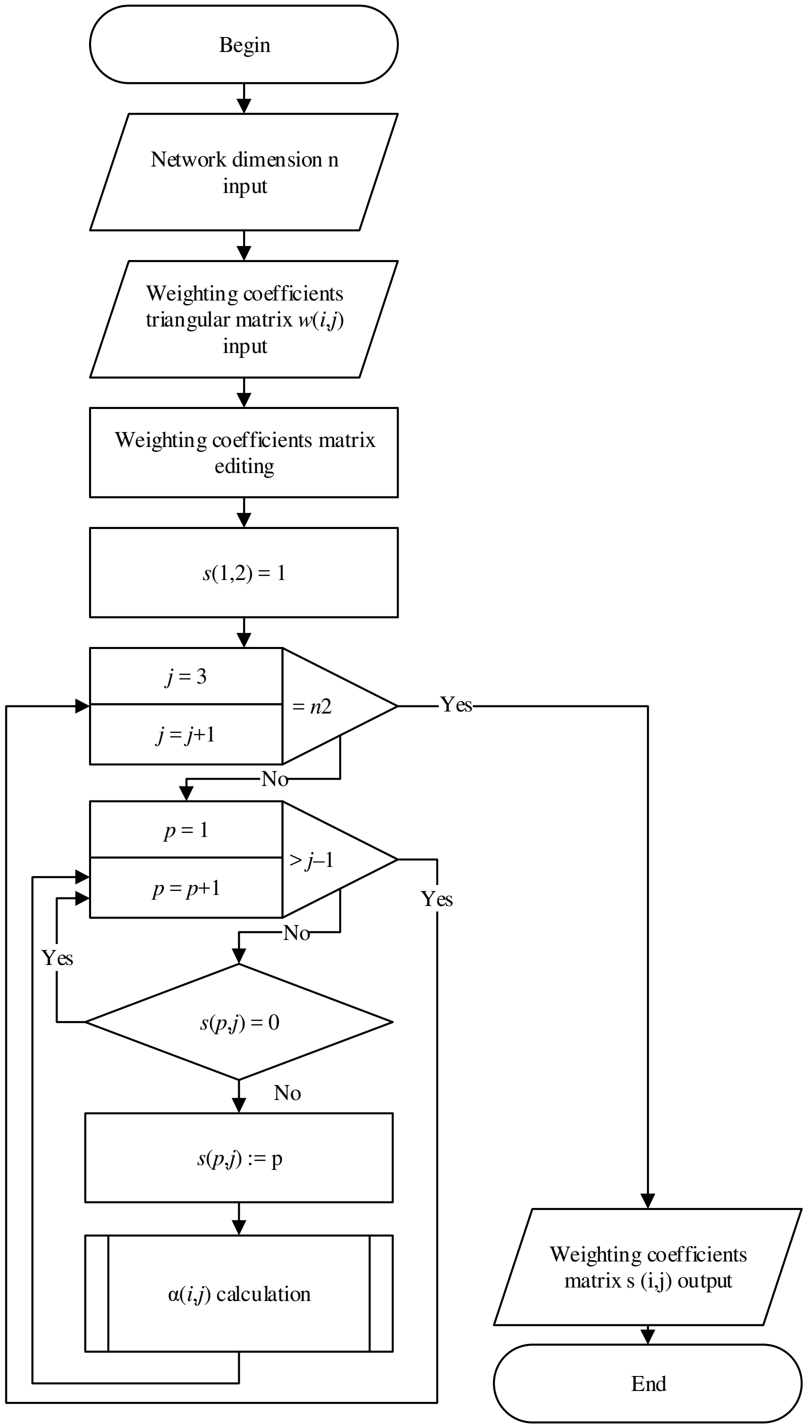

Figure 1. A block diagram of an algorithm for finding the shortest route.

Algorithm. Entrance: *P*– the task of creating schedules. Exit: an optimal for *P* shedule.

```
procedure OPT (P)
begin
if Pempty then return λ (an empty permutation)
else
begin
```

Let *l* be task index with *P*such that H_l is *p*-maximal set for *P*.

$U := H_l - \{l\}$; $V := I(P) - H_l$.

```
return OPT(P/U), , OPT(P/V)
end
end OPT.
```

OPT procedureis recursive, it has ordering task as the only input parameter; he result of its implementation is the optimal permutation in relation to incoming task. Return `<expression>` operator execution is to calculate `<expression>` and return to the point of procedure call with calculated values as a result.

Allocation of information flows in a network that was created by local DSS is assigned to the network management system. Allocation task is to select and establish the optimal information transmission route from information source to consumer considering the situation in the network. [5] shows that the shortest route between the *i*and *j*network nodes is the most reliable.

Let's underlie the modified Floyd algorithm [6] in the basis of routing algorithm. A block diagram of an algorithm is shown in Fig.1.

Let's take a closer look at "Weighting coefficients matrix editing *w(i,j)"* stage.

1. Consider the matrix $a(i, j)$ as its dimension is equal to 2 then:

$a(1,2) = a(2,1) = w(1,2); \quad s(1,2) = s(2,1) = 1$.

2. When kth dimention $a(i, j)$ matrix is constructed, the ($k+1$)thdimention matrix is constructed in the following way:

a) In weighting coefficients matrix $w(i, j)$ revise $k+1$ column and remember weighting coefficients and node numbers that are used to build a route for $k+1$: $s(p, j) = p$;

b) Calculate weighting coefficients of the possible routes $a(i, j) = a(i, p) + w(p, j)$ and select a route with the lowest weight among them;

c) Fill the $k+1$ column and $k+1$ row of the $a(i, j)$ matrix with obtained weighting coefficients.

3. $k := k+2$.

4. If $k = n+1$, then stop, else go to 7.

This algorithm can obtain matrix of the shortest distances between network nodes. This matrix is stored in every local DSS and generated every time the network topology is changed.

To implement data sharing between the local DSS in the process of the DSS functioning it is expedient to use the method of "bulletin board" [7].

While implementing the bulletin board principle, three presentation methods can be used in full:
-facts and conclusions;
- control information that can be divided into knowledge sources and targets;
- the implementation of the bulletin board principle determines the structure of a distributed system.
To select the the information that must be transmitted or received on the local DSS bulletin board, there must be provided appropriate communication strategy. As such strategy it is proposed to use the distributed task solution strategy [3].

Conclusion

The chosen method of identification the prioritized coefficients and a modified algorithm for the route selection allow to reduce time for decision making proccess.

REFERENCES

1. ГУЛАК Н.К., ГЕРАСИМОВ Б.М., ОКСІЮК О.Г.: Принцип побудови й перспективи застосування інтелектуальних систем /Н.К. Гулак, Б.М. Герасимов, О.Г. Оксіюк//Науково-технічна інформація, 2(2006), 48-52.
2. SIMONOVIC A., SLOBODAN P.: Decision suppot for sustainable water resources development in water resources plainning in a changing word /A Simonovic, P. Slobodan // Proceeding of International UNESKO Sumposium, Karlsrune, Germany, III(1994), 3-13.
3. САМОХВАЛОВ Ю.Я.: Декомпозиция логико-лингвистических моделей принятия решений в распределенной вычислительной среде /Ю.Я. Самохвалов// Кибеонетика и системный анализ, 1997, 57-65.
4. КОКОРЕВА Л.В., ПЕРЕВОЗЧИКОВА О.Л. ЮЩЕНКО Е.Л.: Диалоговые системы представления знаний /Л.В. Кокорева, О.Л. Перевозчикова, Е.Л. Ющенко. – Київ: Наукова думка 1992.
5. ЦВИРКУН А.Д.: Основы синтеза структуры сложных систем /А.Д. Цвиркун – Москва: Наука 1982.
6. АРТАМОНОВ Г.Т., ТЮРИН В.Д.: Топология сетей ЭВМ для информационных систем /Г.Т. Артамонов, В.Д. Тюрин – Москва: Радио и связь 1991.
7. Nilp. Blakboard systems.The blackboard model of problem solving. AT Magazine, 17(1986), 38-53.