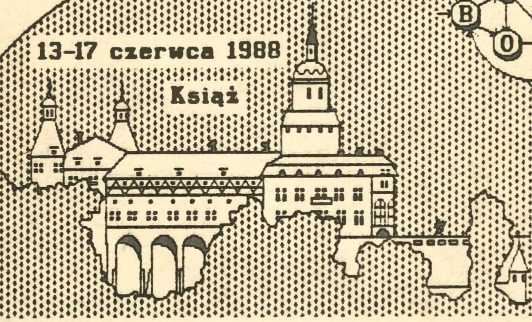
Redaktorzy:
A.Straszak
Z.Nahorski
J.Sikorski

13-17 czerwca 1988

Książ



1. Krajowa Konferencja Badań Operacyjnych i Systemowych

Tom 1

BOS'88

POLSKIE TOWARZYSTWO BADAŃ OPERACYJNYCH I SYSTEMOWYCH

INSTYTUT BADAŃ SYSTEMOWYCH POLSKIEJ AKADEMII NAUK

POLSKIE TOWARZYSTWO BADAŃ OPERACYJNYCH I SYSTEMOWYCH

TOM 1

OPTYMALIZACJA
METODY I ZASTOSOWANIA

I KRAJOWA KONFERENCJA

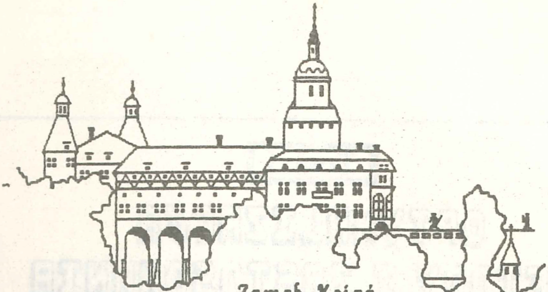
BADAŃ OPERACYJNYCH i SYSTEMOWYCH

Książ, 13 - 17 czerwca 1988

BOS'88

INSTYTUT BADAŃ SYSTEMOWYCH POLSKIEJ AKADEMII NAUK

1989
WARSZAWA



Zamek Książ

I Krajowa Konferencja Badań Operacyjnych i Systemowych

Organizator konferencji

Polskie Towarzystwo Badań Operacyjnych i Systemowych
przy współpracy
Instytutu Badań Systemowych PAN

Komitet naukowy konferencji

Jerzy Hołubiec, Andrzej Kałuszko, Jerzy Kisielnicki, Henryk Kowalowski, Roman Kulikowski, Franciszek Marecki, Zbigniew Nahorski, Stanisław Piasecki, Jarosław Sikorski, Jan Stachowicz, Jan Stasierski, Andrzej Straszak, Maciej Sysło, Władysław Świtalski

Redaktorzy naukowi materiałów

Andrzej Straszak, Zbigniew Nahorski, Jarosław Sikorski

5. Optymalizacja struktur

5.3

I Krajowa Konferencja
Badań Operacyjnych i Systemowych
Książ, 13 - 17 czerwca 1988r.

WYZNACZANIE OPTYMALNEJ STRATEGII TESTOWANIA PROGRAMU

Kazimierz Worwa
Wojskowa Akademia Techniczna
ul. Kaliskiego
01-489 Warszawa

W referacie rozpatruje się zagadnienie wpływu procesu testowania programu na jego niezawodność. W ramach prowadzonych rozważań konstruuje się model matematyczny procesu testowania programu, w ramach którego przedstawia się propozycje wskaźników umożliwiających ocenę wpływu tego procesu na poziom jakości i koszt testowania programu. Dla zilustrowania praktycznej przydatności konstruowamych wskaźników w dalszej części referatu formułuje się zadanie dwukryterialnej optymalizacji strategii testowania programu, z kosztem testowania i wskaźnikiem niezawodności programu jako kryteriami składowymi.

1. Wstęp

Pomimo stałego rozwoju i doskonalenia metod projektowo-implementacyjnych wykorzystywanych w praktyce produkcji oprogramowania, ich aktualny poziom nie daje pełnej gwarancji wytworzenia złożonego produktu programowego całkowicie wolnego od błędów. Błędy te, wykrywane po krótszym lub dłuższym okresie użytkowej eksploatacji oprogramowania, narażają użytkownika na powstawanie określonych, zależnych od charakteru i przeznaczenia tego oprogramowania , strat. Mając na uwadze minimalizację tych strat użytkownik żąda od producenta wytworzenia oprogramowania maksymalnie niezawodnego, a więc dokładnie sprawdzonego i przetestowanego. Z wymienionych powodów proces produkcji produktu programowego kończy się etapem testowania, podstawowym celem którego jest wykrycie i usunięcie maksymalnej liczby popełnionych wcześniej błędów projektowo-programowych.

Etap testowania, stwarzając duże możliwości weryfikacji i kształtowania niezawodności oprogramowania, istotnie podraża jednak koszt jego wytworzenia. Efektywność prac związanych z testowaniem, wyrażająca się wzajemną zależnością poziomu jakości produktu programowego i kosztu jego testowania, silnie zależy od przyjętej strategii testowania, określającej organizację i zakres wykonywanych prac. W opisywanej sytuacji zachodzi zatem potrzeba określenia warunków kompromisu w zakresie stawianych przed oprogramowaniem wymagań jakościowo-kosztowych. Praktyczne znalezienie wspomnianego kompromisu może być znacznie ułatwione w przypadku, gdy istnieją możliwości formalnej oceny poziomu jakości oprogramowania i kosztu jego testowania za pomocą odpowiednich wskaźników.

W referacie przedstawia się propozycje wskaźników umożliwiających ocenę wpływu procesu testowania programu na poziom jego jakości oraz koszt tego procesu. Dla zilustrowania praktycznej przydatności konstruowanych wskaźników w dalszej części referatu formułuje się zadanie dwukryterialnej optymalizacji strategii testowania programu, z kosztem testowania i wskaźnikiem niezawodności programu jako kryteriami składowymi.

2. Opis i model matematyczny procesu testowania programu

Testowany program charakteryzowany będzie za pomocą grafu skierowanego G, określonego następująco:

$$G = (\mathbb{I}, \mathbb{U}),$$

gdzie:

$\mathbb{I}$ - zbiór wierzchołków grafu odpowiadający zbiorowi numerów modułów testowanego programu:

$$\mathbb{I} = \{1, 2, \ldots, i, \ldots, I\},$$

$\mathbb{U} \subset \mathbb{I}\times\mathbb{I}$ - zbiór par uporządkowanych $(i,j)\in\mathbb{I}\times\mathbb{I}$, przy czym para $(i,j)\in\mathbb{U}$, jeżeli po wykonaniu się i-tego modułu (w trakcie wykonywania się programu) jako następny może być wykonywany j-ty moduł.

Bez zmniejszenia ogólności można założyć, że rozpatrywany program ma jeden moduł wejściowy i jeden moduł wyjściowy o numerach i_{WE}, i_{WY} odpowiednio, $i_{WE}, i_{WY} \in I$.

W grafie programu G można wyróżnić pewien zbiór dróg łączących wierzchołek początkowy i_{WE} z wierzchołkiem końcowym i_{WY}, przy czym termin "droga" rozumiany jest jak w teorii grafów i sieci. Z uwagi na to, że rozpatrywany graf G jest grafem skierowanym każda droga łącząca wierzchołki i_{WE}, i_{WY} może być jednoznacznie określona poprzez podanie numerów wierzchołków, przez które "przechodzi". Każda taka droga, dla której ponadto istnieje co najmniej jeden zestaw danych wejściowych programu, który ją uaktywnia nazywana będzie drogą logiczną, przy czym uaktywnienie określonej drogi logicznej oznacza kolejne wykonywanie się modułów wchodzących w jej skład.

Niech Q oznacza zbiór numerów wszystkich dróg logicznych testowanego programu:

$$Q = \{1,2,\ldots,q,\ldots,Q\}.$$

Niech I_q oznacza zbiór numerów q-tej drogi logicznej:

$$I_q = \{i_{q,1}, i_{q,2}, \ldots, i_{q,k}, \ldots, i_{q,I}\}, \qquad q \in Q,$$

gdzie:

$i_{q,k}$ - numer k-tego modułu q-tej drogi logicznej (np. w kolejności wykonywania się modułów wchodzących w jej skład),

I_q - liczba modułów tworzących q-tą drogę logiczną.

W dalszych rozważaniach niezawodność rozpatrywanego programu oceniana będzie w oparciu o następujący wskaźnik (Thayer i in. (1978), Worwa (1985)):

$$R = \sum_{q \in Q} d_q \prod_{i \in I_q} R_i , \qquad (1)$$

gdzie:

R_i - wskaźnik niezawodności i-tego modułu, interpretowany jako prawdopodobieństwo jego poprawnego, pojedynczego wykonania się,

d_q - prawdopodobieństwo uaktywnienia q-tej drogi logicznej przez pojedynczy zestaw danych wejściowych programu.

Zgodnie z (1) wskaźnik niezawodności programu R rozumiany jest jako prawdopodobieństwo jego poprawnego wykonania się dla pojedynczego zestawu danych wejściowych. Wartości prawdopodobieństw d_q, $q \in Q$, zależą od charakteru i częstości występowania określonych zestawów danych, dla których rozpatrywany program jest wykonywany. Stanowią one zatem charakterystykę tzw. roboczego środowiska programu i jako takie mają w prowadzonych rozważaniach charakter obiektywny, tj. niezależny od producenta. W związku z powyższym producent programu może wpływać na wartości wskaźnika niezawodności programu R kształtując odpowiednio wartości wskaźników niezawodności modułów R_i, $i \in I$.

Przyjmuje się, że testowanie rozpatrywanego programu polega na niezależnym testowaniu jego modułów składowych, oraz że testowanie i-tego modułu, $i \in I$, składa się z tzw. cykli, każdy z których obejmuje:

1. testowanie modułu dla pewnej liczby uprzednio przygotowanych zestawów danych testowych,
2. ocenę uzyskanych rezultatów oraz lokalizację i usunięcie ewentualnie wykrytych błędów.

Przyjęta organizacja procesu testowania programu odpowiada występującemu w praktyce etapowi tzw. autonomicznego testowania modułów składowych programu, po którym następuje zwykle etap tzw. integracyjnego testowania programu (Myers (1979), Worwa (1985)).

Niech $S = (S_1, S_2, \ldots, S_i, \ldots, S_I)$ oznacza strategię testowania rozpatrywanego programu, przy czym S_i jest strategią testowania i-tego modułu określoną następująco:

$$S_i = (K_i, (L_{i,1}, L_{i,2}, \ldots, L_{i,k}, \ldots, L_{i,K_i})), \tag{2}$$

gdzie:

K_i - liczba cykli testowania i-tego modułu,

$L_{i,k}$ - liczba zestawów danych (testów), w oparciu o które i-ty moduł jest testowany w k-tym cyklu procesu jego testowania, $k = \overline{1, K_i}$, $i \in I$.

Dla oceny wpływu testowania programu na jego niezawodność szczegółowo przeanalizowany zostanie k-ty cykl procesu testowania i-tego modułu, $1 \le k \le K_i$, $i \in I$.

Niech $N_{i,k}(S_i)$ będzie zmienną losową oznaczającą liczbę błędów wykrytych w i-tym module w k-tym cyklu procesu jego testowania, realizowanego według strategii S_i. Z uwagi na przyjęty schemat testowania rozkład prawdopodobieństwa zmiennej losowej

$N_{i,k}(S_i)$ jest rozkładem dwumianowym postaci:

$$(3) \qquad Pr(N_{i,k}(S_i)=n) = \binom{L_{i,k}}{n} p^{n}_{i,k}(S_i)\, r^{L_{i,k}-n}_{i,k}(S_i),$$

$$n \in \{0,1,2,\ldots\},$$

gdzie:

$$(4) \qquad r_{i,k}(S_i) = R_{i,k-1}(S_i), \qquad p_{i,k}(S_i) = 1 - r_{i,k}(S_i),$$

przy czym wielkość $R_{i,k-1}(S_i)$ oznacza wskaźnik niezawodności i-tego modułu w k-tym cyklu procesu jego testowania według strategii S_i.

Zgodnie z przyjętą organizacją cyklu testowania po wykonaniu $L_{i,k}$ testów wykryte w module błędy podlegają lokalizacji i usuwaniu. Usunięcie z i-tego modułu błędów wykrytych w k-tym cyklu testowania prowadzi do zwiększenia jego niezawodności, tj. zwiększenia wartości wskaźnika $R_{i,k-1}(S_i)$:

$$(5) \qquad R_{i,k}(S_i) = R_{i,k-1}(S_i) + \Delta R_{i,k}(S_i), \quad k \in \{1,2,\ldots,K_i\},$$

gdzie $\Delta R_{i,k}(S_i)$ jest "przyrostem" prawdopodobieństwa poprawnego, jednokrotnego wykonania się i-tego modułu, uzyskanym w wyniku wykonania k-tego cyklu procesu jego testowania według strategii S_i.

Przyrost $\Delta R_{i,k}(S_i)$ jest funkcją liczby błędów wykrytych w k-tym cyklu procesu testowania i-tego modułu. Na podstawie przesłanek literaturowych (Misra (1983), Thayer i in. (1978)) przyjmuje się, że ma on postać:

$$\Delta R_{i,k}(S_i) = [1 - R_{i,k-1}(S_i)][1 - e^{-\alpha_i N_{i,k}(S_i)}], \tag{6}$$

gdzie α_i jest parametrem charakteryzującym "podatność" i-tego modułu na wzrost niezawodności (zależną m. in. od jego złożoności).

Z uwagi na to, że wielkość $N_{i,k}(S_i)$ jest zmienną losową, zarówno przyrost $\Delta R_{i,k}(S_i)$, jak i wskaźnik $R_{i,k}(S_i)$ także są wielkościami losowymi. Przyjmując, że $R_{i,o}(S_i) = R_i$, z zależności (5) i (6) otrzymuje się:

$$R_{i,k}(S_i)\Bigg|_{\substack{N_{i,1}(S_i)=n_1 \\ N_{i,2}(S_i)=n_2 \\ \vdots \\ N_{i,k}(S_i)=n_k}} = 1 - e^{-\alpha_i \sum_{l=1}^{k} n_l}(1 - R_i). \tag{7}$$

Zależność (7), umożliwiająca wyznaczenie wartości wskaźnika niezawodności i-tego modułu $R_{i,k}(S_i)$, po zakończeniu k-tego cyklu procesu jego testowania według strategii S_i, ma charakter warunkowy. Bezwarunkowa wartość tego wskaźnika określona jest następująco:

$$R_{i,k}(S_i) = \sum_{n_1=0}^{L_{i,1}} \sum_{n_2=0}^{L_{i,2}} \cdots \sum_{n_k=0}^{L_{i,k}} [\, R_{i,k}(S_i)\Bigg|_{\substack{N_{i,1}(S_i)=n_1 \\ N_{i,2}(S_i)=n_2 \\ \vdots \\ N_{i,k}(S_i)=n_k}} \cdot Pr\{N_{i,1}(S_i)=n_1,\, N_{i,2}(S_i)=n_2,\, \ldots,\, N_{i,k}(S_i)=n_k\}\,]. \tag{8}$$

Ponieważ

$$(9) \quad Pr\{N_{i,1}(S_i)=n_1,\ N_{i,2}(S_i)=n_2,\ \ldots,\ N_{i,k}(S_i)=n_k\} =$$

$$= \prod_{l=1}^{k} Pr\{N_{i,l}(S_i)=n_l \,\Big|\, \begin{matrix} N_{i,1}(S_i)=n_1 \\ N_{i,2}(S_i)=n_2 \\ \vdots \\ N_{i,l-1}(S_i)=n_{l-1} \end{matrix}\},$$

więc po uwzględnieniu zależności (3) i (6) wyrażenie określające wartość wskaźnika $R_{i,k}(S_i)$ można przedstawić w postaci:

$$(10) \quad R_{i,k}(S_i) = \sum_{n_1=0}^{L_{i,1}} \sum_{n_2=0}^{L_{i,2}} \ldots \sum_{n_k=0}^{L_{i,k}} \{[1 - e^{-\alpha_i \sum_{l=1}^{k} n_l}(1 - R_i)] \cdot$$

$$\cdot \prod_{l=1}^{k} \binom{L_{i,l}}{n_l} p_{i,l}^{n_l}(S_i)\, r_{i,l}^{L_{i,l}-n_l}(S_i)\},\ k\in\{1,2,\ldots,K_i\},\ i\in I,$$

gdzie prawdopodobieństwa $p_{i,l}(S_i)$, $r_{i,l}(S_i)$ określone są zgodnie z (4).

Realizacja k-tego cyklu procesu testowania i-tego modułu według strategii S_i wiąże się z określonym kosztem. Koszt ten, oznaczany dalej przez $C_{i,k}(S_i)$, obejmuje koszt przygotowania i komputerowego przetworzenia (przez testowany program) wykorzystywanych zestawów danych oraz koszt lokalizacji i usunięcia ewentualnie wykrytych błędów. Koszt $C_{i,k}(S_i)$ można określić następująco:

$$(11) \quad C_{i,k}(S_i) = L_{i,k}\cdot\bar{t}_i + N_{i,k}(S_i)\cdot\bar{u}_i,\quad k\in\{1,2,\ldots,K_i\},\ i\in I,$$

gdzie:

- $\bar{t}_i$ - średni koszt przygotowania i komputerowego przetworzenia jednego zestawu danych dla i-tego modułu,
- $\bar{u}_i$ - średni koszt usunięcia z i-tego modułu jednego błędu.

Z uwagi na to, że liczba błędów $N_{i,k}(S_i)$, wykrytych w k-tym cyklu procesu testowania i-tego modułu, jest zmienną losową, koszt $C_{i,k}(S_i)$ również jest zmienną losową.

3. Sformułowanie zadania optymalizacji strategii testowania programu

Zgodnie z rozpatrywanym schematem testowania wskaźnik niezawodności programu $R(S)$ po zakończeniu jego testowania według strategii S można określić następująco:

$$(12) \qquad R(S) = \sum_{q \in Q} d_q \prod_{i \in I_q} R_i(S_i),$$

gdzie wielkość $R_i(S_i)$ jest wskaźnikiem niezawodności i-tego modułu po zakończeniu procesu jego testowania według strategii S_i.

Ponieważ $R_i(S_i) = R_{i,K_i}(S_i)$, więc z zależności (10) otrzymuje się:

$$(13) \qquad R_i(S_i) = \sum_{n_1=0}^{L_{i,1}} \sum_{n_2=0}^{L_{i,2}} \dots \sum_{n_{K_i}=0}^{L_{i,K_i}} \{[1 - e^{-\alpha_i \sum_{l=1}^{K_i} n_l} (1 - R_i)] \cdot$$

$$\cdot \prod_{l=1}^{K_i} \binom{L_{i,l}}{n_l} p_{i,l}^{n_l}(S_i)\, r_{i,l}^{L_{i,l}-n_l}(S_i)\}, \quad i \in I.$$

Niech $C(S)$ oznacza koszt procesu testowania rozpatrywanego programu, realizowanego według strategii S. Koszt $C(S)$ można określić następująco:

$$(14) \qquad C(S) = \sum_{i \in I} \sum_{k=0}^{L_{i,K_i}} C_{i,k}(S_i),$$

gdzie $C_{i,k}(S_i)$ jest kosztem k-tego cyklu procesu testowania i-tego modułu według strategii S_i, określonym zależnością (11).

Zgodnie z wcześniejszymi uwagami koszt $C(S)$ jest zmienną losową. W dalszych rozważaniach wykorzystywana będzie wartość oczekiwana kosztu $C(S)$, która - zgodnie z zależnościami (10) i (13) - wyraża się wzorem:

$$(15) \qquad E_n[C(S)] = \sum_{i \in I} \sum_{k=0}^{L_{i,K_i}} L_{i,k} [\bar{t}_i + (1-R_{i,k-1}(S_i)) \cdot \bar{u}_i].$$

Ze względów praktycznych na wskaźnik niezawodności $R(S)$ i koszt $E[C(S)]$ nakłada się następujące ograniczenia:

$$(16) \qquad \begin{aligned} R(S) &\geq R_{MIN}, \\ E[C(S)] &\leq C_{MAX}, \end{aligned}$$

gdzie wielkości R_{MIN}, C_{MAX} oznaczają odpowiednio minimalny, dopuszczalny poziom niezawodności testowanego programu i maksymalny, dopuszczalny poziom kosztu ponoszonego z tytułu testowania.

W oparciu o wprowadzone oznaczenia oraz uzyskane zależności sformułować można następujące dwukryterialne zadanie optymalizacji strategii testowania programu:

$$(17) \qquad (\mathcal{S}, F, \Phi),$$

gdzie:

$\mathcal{S}$ - zbiór rozwiązań (strategii) dopuszczalnych, określony następująco:

$\mathcal{S} = \{ S=(S_1, S_2, ..., S_i, ..., S_I) : S_i$ jest określone zależnością (2) i spełnione są ograniczenia (16) $\}$,

F - wskaźnik jakości rozwiązania postaci:

$$F(S) = (R(S), E[C(S)]),$$

przy czym wielkości $R(S)$, $E[C(S)]$ są określone zależnościami (12) i (15) odpowiednio,

Φ - relacja dominowania określona następująco:

$$\Phi = \{ (y_1, y_2) \in Y \times Y : y_1^1 \geq y_2^1, y_1^2 \leq y_2^2 \},$$

$$y_1 = (y_1^1, y_1^2), \quad y_2 = (y_2^1, y_2^2),$$

gdzie Y jest tzw. przestrzenią kryterialną, określoną gdzie $\mathbb{Y}$ jest tzw. przestrzenią kryterialną, określoną jak niżej:

$$\mathbb{Y} = \{ y=(R(S), E[C(S)]): S \in \mathbb{S} \}.$$

Zadanie (17) jest nieliniowym, całkowito-liczbowym zadaniem optymalizacji dwukryterialnej. Jego rozwiązanie może być wyznaczone zgodnie z ogólnie przyjętą metodyką rozwiązywania zadań polioptymalizacji (Ameljańczyk (1986)).

4. Uwagi końcowe

W przedstawionych rozważaniach skoncentrowano się na wybranych aspektach określania strategii testowania poszczególnych modułów składowych programu. Ograniczając się do problemu oceny wpływu liczby zestawów danych testowych na niezawodność modułów pominięto zagadnienie ich szczegółowego doboru. Podejście takie jest uzasadnione w przypadku, gdy z uwagi na specyfikę modułów, kolejne zestawy danych testowych mogą być losowane (bez zwracania) spośród zbioru wszystkich możliwych zestawów danych poszczególnych modułów.

Analizę wpływu testowania modułu na jego niezawodność przeprowadzono przy założeniu, że w ramach jego jednego wykonania się może zostać wykryty co najwyżej jeden błąd. Z uwagi na to, że w rzeczywistości liczba wykrywanych w ten sposób błędów może być większa, zależności określające przyrosty niezawodności poszczególnych modułów dają oszacowania pesymistyczne.

Przedstawiony model matematyczny procesu testowania programu może być przydatny w pracach mających na celu badanie efektywności testowania oprogramowania, formułowanie zadań optymalizacyjnych z zakresu planowania i organizacji testowania itp.

Literatura

1. Ameljańczyk A. (1986) Optymalizacja wielokryterialna. WAT.
2. Misra P.N. (1983) Software reliability analysis. IBM Systems Journal, Vol. 22, N° 3.
3. Myers G.J. (1979) The art of software testing. Wiley-

Interscience Publication.
4. Thayer T.A., Lipov M., Nelson E.C. (1978) Software reliability. North-Holland Publishing Company.
5. Worwa K. (1985) Analiza wpływu testowania integracyjnego na niezawodność programu. VIII International Conference FTSD'85. Proceedings.

L I T E R A T U R A

(1) Crabill T. B., Gross D., Magazine M. J. (1977) A Classified Bibliography of Research on Optimal Design and Control of Queues. Operations Res. Vol. 25, No. 2, pp. 219-232.

(2) Deb R. K., Serfozo R. F. (1973) Optimal Control of Batch Service Queues. Advances Appl. Probability 5, 340-361.

(3) [illegible] (1979) [illegible] Theory of Service [illegible]. Pergamon Press, New York [illegible]

(4) [illegible] (1979) The Computation of Optimal Control Limits for a Queue with Batch Services, Manag. Sci. 25 [illegible]

(5) [illegible] (1983) [illegible]. Praca doktorska, [illegible]

(6) Королюк В., Турбин А. (1976) Полумарковские процессы и их приложения. Киев [illegible]

(7) [illegible] (1982) [illegible]

(8) Neuts M. (1967) A general class of bulk queues with Poisson input. Ann. Math. Stat. 38, 759-770.

(9) ______ (1979) Queues solvable without Rouché's theorem. Operations Res. 27.

(10) Keilson J. (1966) The ergodic queue length distributions for queueing systems with finite capacity. J. Roy. Stat. Soc. B28.

Zarząd

Polskiego Towarzystwa Badań Operacyjnych i Systemowych

Prezes

prof.dr hab.inż. Andrzej Straszak
Instytut Badań Systemowych PAN

Wiceprezes

prof.dr hab.inż. Jan Stasierski
Wojskowa Akademia Techniczna

Wiceprezes

prof.dr hab.inż. Stanisław Piasecki
Instytut Badań Systemowych PAN

Sekretarz generalny

dr inż. Zbigniew Nahorski
Instytut Badań Systemowych PAN

Sekretarz

dr inż. Jarosław Sikorski
Instytut Badań Systemowych PAN

Skarbnik

dr inż. Andrzej Kałuszko
Instytut Badań Systemowych PAN

Członkowie

prof.dr hab. Jerzy Kisielnicki
Wydział Zarządzania UW

doc.dr hab.inż. Bohdan Korzan
Wojskowa Akademia Techniczna

doc.dr hab.inż. Jan Stachowicz
Zakład Nauk Zarządzania PAN

doc.dr hab.inż. Maciej Sysło
Instytut Informatyki UWr.

Komisja rewizyjna

PRZEWODNICZĄCY

dr Władysław Świtalski
Katedra Cybernetyki i Badań Operacyjnych UW

CZŁONKOWIE

dr inż. Janusz Kacprzyk
Instytut Badań Systemowych PAN

dr inż. Marek Malarski
Instytut Transportu PW

doc.dr hab.Henryk Sroka
Akademia Ekonomiczna w Katowicach

dr inż. Leon Słomiński
Instytut Badań Systemowych PAN

PION III