

O PROJEKTOWANIU ODPORNYM NA NIEDOSKONAŁOŚCI MODELU NUMERYCZNEGO

Mariusz KAMOLA*†

*Politechnika Warszawska, Instytut Automatyki i Informatyki Stosowanej,
ul. Nowowiejska 15/19, 00-665 Warszawa, e-mail: mkamola@ia.pw.edu.pl

†Naukowa i Akademicka Sieć Komputerowa,
ul. Wąwozowa 18, 02-796 Warszawa, e-mail: Mariusz.Kamola@nask.pl

Streszczenie: Procesy fizyczne określające zachowanie istniejących i projektowanych obiektów często muszą być modelowane w sposób nieanalityczny, poprzez rozwiązanie zadania numerycznego. Samo skomplikowanie modelu jak i niedoskonałości takiego modelowania skutecznie utrudniają automatyzację zadania doboru wartości zmiennych decyzyjnych obiektu w celu uzyskania jego optymalnego zachowania. W pracy wskazuje się cechy wspólne tak postawionych zadań optymalizacji (model numeryczny + algorytm optymalizacji) i proponuje pewną uniwersalną procedurę projektową mającą na celu skonstruowanie i dostosowanie algorytmu optymalizacji do konkretnej klasy problemów, reprezentowanych przez model. Analizuje się skuteczność tejże procedury na dwóch przykładach praktycznych. Rozważa się możliwość zautomatyzowania procedury projektowej; w tym celu proponowane są założenia i ogólny kształt docelowego uniwersalnego algorytmu optymalizacji.

Słowa kluczowe: komputerowo wspomagane projektowanie, modele numeryczne, hybrydowe algorytmy optymalizacji

1. PRZEDSTAWIENIE PROBLEMU

Rozważmy zadania optymalizacji, w których każdorazowo do wyznaczenia wartości funkcji celu potrzeba rozwiązać pewne zadanie numeryczne. Konkretnymi przykładami takich zadań mogą być problemy komputerowo wspomaganego projektowania różnych systemów (urządzeń, zakładów produkcyjnych itp.) — wówczas przedmiot modelowania i optymalizacji jeszcze nie istnieje. Inne przykłady mogą dotyczyć doboru optymalnych parametrów pracy obiektów istniejących fizycznie, ale — z różnych względów — zamodelowanych matematycznie i dopiero w takiej postaci będących przedmiotami optymalizacji. Oczywiście, w obu rozważanych grupach owo zadanie numeryczne polega właśnie na “rozwiązaniu” modelu, tj. wyznaczeniu wartości elementów wektora zmiennych zależnych (wyjściowych modelu) y przy danym wektorze x zmiennych decyzyjnych (wejściowych). Zadanie optymalizacji polega na szukaniu x^*

minimalizującego wskaźnik jakości $f(x, y)$

$$\min_{x,y} f(x, y) . \quad (1)$$

Patrząc od strony funkcjonalnej, dostrzega się więc dwuczłonową strukturę rozważanego zadania optymalizacji: oto moduł nadrzędny, optymalizacji, zleca wykonanie zadania numerycznego modułowi podrzędnemu, symulatorowi. Przypomina ona strukturę optymalizacji hierarchicznej metodą bezpośrednią [2, s. 36–37], gdzie odpowiednikiem symulacji jest lokalne zadanie optymalizacji, a odpowiednikiem optymalizacji — zadanie koordynacji. (Również trudności stwarzane przez takie układy są w obu przypadkach podobne.)

Ograniczenia dla zadania optymalizacji (1) są następujące:

$$h(x, y) = 0 , \quad (2)$$

$$x \in D_x , \quad y \in D_y , \quad (3)$$

gdzie (2) oznacza rozwiązanie modelu numerycznego, który w sposób uwikłany opisuje zależności między x a y , a (3) opisuje dodatkowe ograniczenia kostkowe nałożone zarówno na x jak i na y . Fakt, że wyznaczenie y z x odbywa się w sposób numeryczny — przez rozwiązanie (2) — ma ważkie następstwa. Przez “sposób numeryczny” nie rozumie się bowiem zaimplementowania i wyliczenia jakiegoś wzoru, ale zastosowanie procedury obliczeniowej nie mającej spójnych podstaw analitycznych. Przykładem może być iteracyjne znajdowanie miejsc zerowych funkcji nieliniowej. Bezpośrednim utrudnieniem wynikającym ze stosowania takiej procedury jest niemożność rozwikłania zależności $h(\cdot)$ wiążącej y z x . Ponadto, przez dyskretną naturę obliczeń komputerowych, wynik jest z reguły obciążony “szumem” numerycznym: powtarzalnymi ale nieprzewidywalnymi fluktuacjami. Oprócz dwóch powyższych niedogodności obserwuje się i inne, mocniej związane z nieprzebraniem reguł sztuki modelowania i implementacji: przełączaniem wewnątrz modelu i awariami symulatora.

Przełączanie objawia się np. nagłymi skokami y dla łągodnie zmieniającego się x i może być w szczególności

wynikiem stosowania modeli obszarowych, nieumiejętnie połączonych. Natomiast awarie symulatora zazwyczaj wynikają z nie dość solidnego zaimplementowania algorytmu rozwiązującego model; powszechnym objawem jest brutalne przerwanie obliczeń i zgłoszenie sytuacji wyjątkowej przez system przy próbie wykonania operacji o argumentach leżących poza jej dziedziną. Awaria symulatora w zasadzie oznacza niemożność znalezienia takiego y dla danego x , by (2) było spełnione; jednakże mocno komplikuje sytuację, bo nie pozostawia żadnego wyniku, żadnej informacji nawet o tym, w jakim stopniu (2) zostało naruszone.

Należy zauważyć, że wszystkie powyższe cechy modeli numerycznych mogą znajdować dobre uzasadnienie: iteracyjne poszukiwanie miejsc zerowych bywa jedyną skuteczną techniką, szum numeryczny jest nieodłącznym efektem dyskretyzacji zachodzącej w komputerze, przełączanie może być immanentną cechą obiektu i odzwierciedlać np. “katastrofy” (nagłe jakościowe zmiany stanu), wreszcie przerwanie obliczeń bywa odpowiednikiem rzeczywistej akcji jak np. awaryjnego wyłączenia systemu i opisuje stan systemu o wiele trafniej niż gdyby np. miało wylizywać zupełnie absurdalne wartości y . Zatem model, jakkolwiek trudny, jest niepodatny na dalsze uproszczenia, które umożliwiałyby użycie sprawnych algorytmów optymalizacji. W takich sytuacjach nie można wymagać ani rozwikłania modelu (znalezienia jawnej zależności y od x), co się powszechnie postuluje [4, s. 383], ani innych zabiegów upraszczających problem (eliminacja niektórych ograniczeń). Warto również dodać, że symulator jest przeważnie zamkniętym modułem obliczeniowym którego działania i tak nie sposób już zmienić.

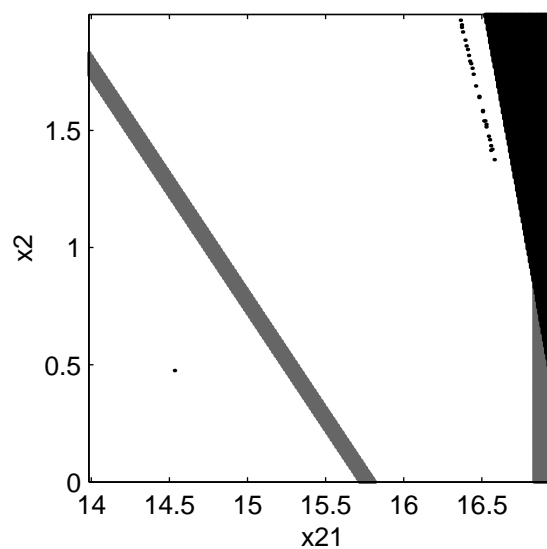
Pozostaje zatem zmierzyć się z problemem optymalizacji cech obiektu reprezentowanego przez kłopotliwy symulator — z wszystkimi niedogodnościami. Nie jest to wątek i podejście szeroko reprezentowane w literaturze, mimo powszechnej opinii o częstym występowaniu zadań optymalizacji zdefiniowanych w ten sposób [5, s. 177]. Niemniej jednak, z uwagi na częste cechy wspólne takich problemów, jest możliwe opracowanie uniwersalnej procedury projektowej, prowadzącej do opracowania algorytmu optymalizacji skutecznego i odpornego w obliczu konkretnego problemu. Prace nad powyższym zagadnieniem stały się przedmiotem rozprawy doktorskiej autora. W dalszym ciągu tego artykułu zostaną przedstawione rozpatrywane praktyczne problemy optymalizacji (rozdz. 2.), zaproponowana postać uniwersalnej procedury projektowej (rozdz. 3.) i wyniki jej zastosowania do problemów praktycznych (rozdz. 4.). Artykuł kończy podsumowanie i wskazanie kierunku dalszych badań (rozdz. 5.).

2. ROZPATRYWANE PRAKTYCZNE ZADANIA OPTIMALIZACJI

Przyjrzyjmy się dwóm zadaniom optymalizacji, obu praktycznym.

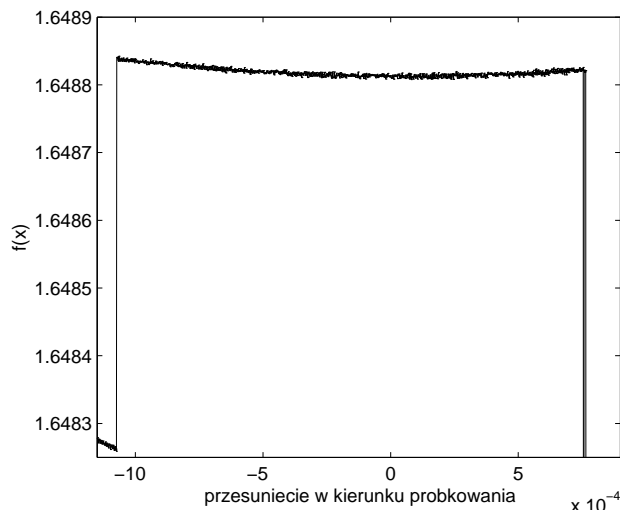
2.1. Optymalizacja punktu pracy elektrociepłowni

Pierwsze polega na wyznaczeniu optymalnego punktu pracy (zapisywanego w postaci wektora temperatur, przepływów, ciśnień i entalpii) dla istniejącej elektrociepłowni przykładowej. Jakość rozwiązania jest określana kryteriami ekonomicznymi, tj. przez bilans kosztów i przychodów. Z przyczyn oczywistych w optymalizacji do określenia punktu pracy (i wynikającej z niego jakości rozwiązania) używa się modelu numerycznego układu [7]. W procesie symulacji działania elektrociepłowni z kilkunastu zmiennych decyzyjnych x jest wyznaczanych kilkaset zmiennych zależnych y , niezbędnych do obliczenia $f(x, y)$. Model elektrowni wykazuje wszystkie niekorzystne cechy wyszczególnione w rozdz. 1. W szczególności nie jest możliwe wyznaczenie czy wyestymowanie pochodnych zmiennych zależnych względem zmiennych decyzyjnych — między innymi z powodu przełączeń dokonywanych w modelu. Ponadto, próba modelowania zachowania obiektu skutkuje awarią symulatora dla stosunkowo dużego podzbioru obszaru zmienności zmiennych decyzyjnych. Ostatecznie



Rys. 1. Przykładowy przekrój przez obszar zmienności x z zaznaczonymi obszarami, w których spełnione są wszystkie ograniczenia (kolor biały), w których niespełnione jest $y \in D_y$ (szary), i w którym niespełnione jest (2), czyli zgłaszany jest wyjątek (czarny).

zbiór dopuszczalnych punktów pracy jest niespójny (por. rys. 1), a dodatkowo funkcja celu wykazuje schodkowość i wpływy szumów numerycznych (por. rys. 2). Szukanie optimum w zadaniu o takich własnościach jest niezwykle trudne — dotychczas było wykonywane metodą

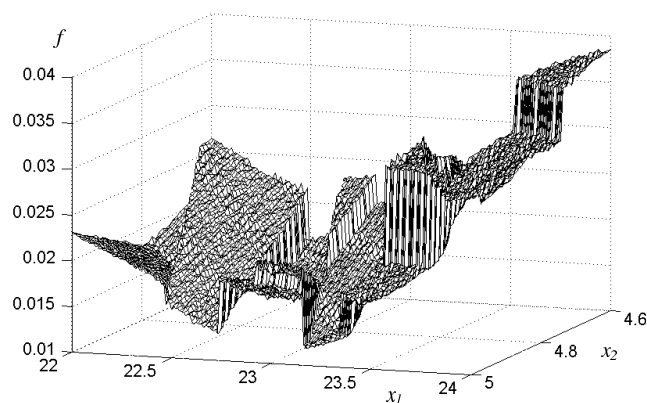


Rys. 2. Przykładowy wykres kierunkowy funkcji celu; widoczne efekty szumów numerycznych i przełączeń wewnątrz modelu.

prób i błędów przez osobę o dużej wiedzy o sterowanym obiekcie.

2.2. Projektowanie falowodu

Drugie zadanie praktyczne polega na zaprojektowaniu falowodu o określonych, optymalnych parametrach. Zachowanie nieistniejącego jeszcze urządzenia jest modelowane przez uniwersalny symulator propagacji pola elektromagnetycznego [6]. W tym przykładzie dominującym utrudnieniem są szumy numeryczne i przełączanie wewnątrz symulatora. Funkcja celu jest



Rys. 3. Wykres funkcji celu dla zadania projektowania falowodu.

więc nieciągła, o wielu minimach lokalnych (por. rys. 3).

Należy podkreślić iż poprawienie tych dwóch modeli tak, aby usunąć opisywane niedogodności jest niemożliwe, i to nie tylko dlatego, że taka jest natura zaimplementowanych w nich algorytmów. Istotną przeszkodą bywa

także fakt, że są one bardzo skomplikowane (model elektrociepłowni) albo wręcz utajnione (symulator pola).

3. IDEA ROZWIĄZYWANIA I PROPONOWANA PROCEDURA PROJEKTOWA

Powyższe dwa przykłady mają na celu uświadomienie, jak różnorodne mogą być kluczowe przeszkody w użyciu typowych, uznanych, zaawansowanych metod optymalizacji stosowanych w przemyśle (GRG, SQP). Asortyment utrudnień jest z pewnością dużo większy i dlatego warto zadać sobie pytanie, czy możliwe jest opracowanie pewnej uniwersalnej procedury projektowej — zbioru zaleceń, których przestrzeganie w obliczu konkretnego problemu prowadzi do skonstruowania skutecznego i wydajnego algorytmu optymalizacji. Z pewnością procedura taka powinna dopuszczać stosowanie różnych metod, w miarę postępu optymalizacji jak i regresu w przypadku wykrycia nowych, niekorzystnych cech problemu. W ostateczności więc powstanie algorytm hybrydowy.

Proponuje się następującą procedurę rozwiązywania zadania:

1. Posiłkując się całą dostępną wiedzą o problemie, wytypować zbiór metod optymalizacji, które mogłyby wejść w skład algorytmu hybrydowego, tj. polegającego na sekwencyjnym aktywowaniu metod w różnych, odpowiednich fazach procesu poszukiwania optimum.
2. Sformułować zadanie uproszczone (np. o zmniejszonej wymiarowości), ale zachowujące — z punktu widzenia optymalizatora — wszystkie cechy jakościowe zadania pierwotnego. Jest to potrzebne do zwiększenia szybkości obliczeń na etapie doboru parametrów algorytmu hybrydowego.
3. Uruchomić algorytm hybrydowy tak, aby uzyskać dość dokładne rozwiązanie zadania uproszczonego (kosztem dużego nakładu obliczeniowego) mogące służyć za punkt odniesienia. W trakcie pracy algorytmu zwrócić uwagę, czy osiągnięte rozwiązanie jest zdroworoządkowe i jakie mogą być przyczyny ewentualnego niepowodzenia optymalizacji: dobór niewłaściwych metod składowych, niewłaściwych parametrów uruchomienia, niewielkie ale brzemienne w skutkach niedostosowanie metody do specyfiki problemu. Stosownie zareagować: wymienić metodę, zmienić parametry, dokonać adaptacji metody.
4. Ustalić kryteria przełączeń między metodami składającymi się na algorytm hybrydowy tak, aby w pierwszym rzędzie zapewnić skuteczność, a następnie — wydajność optymalizacji. Tak zaprojektowany i dostrojony algorytm stosować do rozwiązywania zadania pierwotnego.

Niektóre zalecenia w powyższej procedurze są podobne do rozwiązań stosowanych przez innych autorów, albo wręcz powszechnych [1]. Takim jest założenie o hybrydowej postaci algorytmu i o tym, że wystarczy rozważyć niewielki zbiór metod mogących potencjalnie wejść w skład takiej hybrydy. Z kolei zalecenie skonstruowania modelu uproszczonego przywodzi na myśl zarówno techniki powszechne, np. RSM (Response Surface Methodology), jak i niszowe (stosowanie modeli ciągłych dla procesów dyskretnych, modelowanie za pomocą sztucznych sieci neuronowych itp.). Wszystkie mają na celu przyspieszenie jakiejś fazy rozwiązywania zadania. Uproszczenie proponowane tutaj miałoby zachowywać charakter rozwiązania i dokładność symulacji kosztem redukcji liczby zmiennych decyzyjnych (a nie — jak się powszechnie proponuje — dokładności modelowania).

4. PROCEDURA PROJEKTOWA W DZIAŁANIU

Procedurą wyłożoną w rozdz. 3. posłużono się w celu rozwiązania zadań praktycznych z rozdz. 2. [3]. Właściwie, to definiowanie tejże procedury następowało równoległe z rozwiązywaniem zadań praktycznych gdyż zabiegi stosowane w obu przypadkach w celu skonstruowania sprawnego algorytmu były wystarczająco podobne, aby można było pokusić się o uogólnienie.

4.1. Optymalizacja punktu pracy elektrociepłowni

Początkowy stan wiedzy o problemie optymalizacji punktu pracy elektrociepłowni uzasadniał wytypowanie metody sterowanego przeszukiwania losowego (CRS2) w początkowej fazie i uogólnionego gradientu zredukowanego (GRG) w końcowej fazie optymalizacji. Wdrożono CRS2 dla zwiększenia prawdopodobieństwa wykrycia obszaru atrakcji optimum globalnego, a GRG dla zapewnienia szybkiej zbieżności w tym obszarze i to bez naruszania ograniczeń podczas poszukiwania minimum. Jednakże — już dla modelu uproszczonego — okazało się, że wiarygodna estymacja gradientu funkcji celu względem zmiennych decyzyjnych nie jest możliwa, więc zamiast GRG zastosowano metodę sterowanego poszukiwania sympleksowego (COMPLEX). Wcale nierzadkie (ok. 10% punktów dla których rozwiązywano model) przypadki awarii symulatora stały się istotną przeszkodą we wdrożeniu zarówno CRS2 jak i COMPLEX w ich wersji oryginalnej. Dopiero po wprowadzeniu dość istotnych poprawek uzyskano istotne zwiększenie niezawodności algorytmu hybrydowego. Zatem ostateczny algorytm aktywuje początkowo CRS2, a następnie COMPLEX, przy czym obie metody zostały usprawnione na potrzeby zadania. Próby użycia strategii ewolucyjnej w miejscu CRS2 nie przyniosły istotnych zmian skuteczności i wydajności.

W tabeli 1 zestawiono wyniki uzyskiwane przez algorytm hybrydowy bez ulepszeń, algorytm z ulepszeniami oraz

Table 1. Jakość i koszt rozwiązań uzyskanych przez różne narzędzia optymalizacji.

Algorytm	Hybrydowy autora		Używany przez Epogy
	COMPLEX oryginalny	COMPLEX z ulepszeniami	
wartość średnia $f(\mathbf{x}^*)$	1,738	1,658	1,650
całk. nakład obliczeniowy	45 000	90 000	45 000

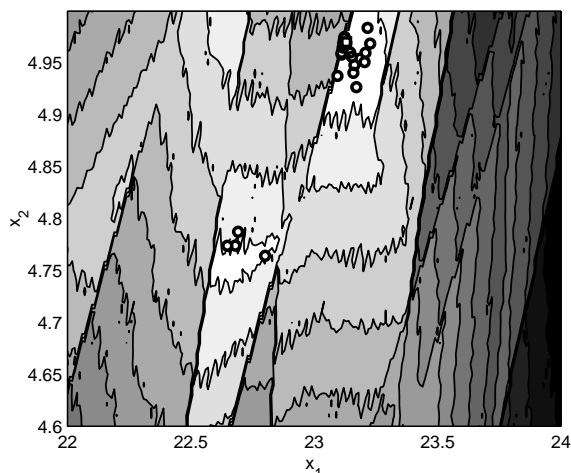
komercyjne narzędzie Epogy, rozwiązujące zadania rozważanego przez nas typu. Algorytm hybrydowy uruchomiono 30-krotnie, Epogy — jednokrotnie. Średnia wartość uzyskiwanych rozwiązań poprawia się dla kolejnych rozpatrywanych algorytmów. Widać korzyści z wprowadzonych usprawnień (aczkolwiek Epogy wciąż okazuje się najskuteczniejszym narzędziem). Analiza nakładu obliczeniowego wskazuje, że za poprawę jakości rozwiązań uzyskiwanych przez usprawniony COMPLEX trzeba zapłacić znacznie większym nakładem obliczeniowym. Jednakowoż, usprawnienia COMPLEX umożliwiają znajdowanie pojedynczych rozwiązań o wiele lepszych niż najlepsze znajdowane przez COMPLEX bez usprawnień.

Osobnym problemem jest dobór właściwego warunku przełączenia się z metody wstępnej na końcową. Skrupulatne testy wykazały, że optymalizowanie metodą CRS2 (lub inną) tylko tak długo, jak jest to konieczne (tj. do uzyskania pierwszego rozwiązania dopuszczalnego, potrzebnego metodzie COMPLEX) jest najlepszym wyjściem. Użycie takiego właśnie kryterium przełączenia daje działanie satysfakcjonujące również w zadaniu pierwotnym: zadanie jest rozwiązywane w czasie akceptowalnym, a uzyskiwane rozwiązanie jest — w opinii ekspertów z dziedziny — satysfakcjonujące. Mimo, że wydajność algorytmu hybrydowego w wersji ostatecznej jest niższa od Epogy, to jego stosowanie ma tę zaletę, że pozwala w pełni kontrolować i dowolnie zmieniać zachowanie metod składowych na miarę wymagań zadania.

4.2. Projektowanie falowodu

Odmienne niż w przypadku optymalizacji punktu pracy elektrociepłowni, zadanie projektowania falowodu było już wcześniej zautomatyzowane — poprzez sprzężenie symulatora z metodą optymalizacji Powella. Jednakże duża zawodność takiego rozwiązania (a konkretnie silne uzależnienie uzyskiwanego rozwiązania od położenia punktu startowego) stała się motorem poszukiwań jakiegoś lepszego algorytmu optymalizacji.

Do konstrukcji algorytmu zastosowano ponownie procedurę projektową przedstawioną w rozdz. 3. Wytypowano metodę CRS2 do uruchomienia w fazie wstępnej i utrzymano dotychczasową metodę Powella w końcowej fazie optymalizacji. Położenie przykładowych rozwiązań znalezionych przez ten algorytm hybrydowy oraz przykła-



Rys. 4. Lokalizacja rozwiązań zadania projektowania falowodu. Kwadrat oznacza położenie wyjątkowo dobrego rozwiązania znalezione wyłącznie metodą Powella. Koła oznaczają położenie typowych rozwiązań znajdowanych przez algorytm hybrydowy.

dowego, dobrego jakościowo rozwiązania znalezione samotnie przez algorytm Powella ilustruje rys. 4. Większość rozwiązań skupia się w obszarze atrakcji minimum globalnego, pozostałe — w okolicy minimum lokalnego. Należy zaznaczyć, że znakomita większość rozwiązań znajdowanych samotnie przez metodę Powella (uruchamianą, podobnie jak algorytm hybrydowy, z losowego punktu początkowego) była bardzo złej jakości i nie została uwidoczniiona na rysunku.

Kwestia doboru chwili przełączenia z CRS2 na algorytm Powella nabiera w tym zadaniu znowu znaczenia: już nie wystarczy przełączyć się po znalezieniu pierwszego punktu dopuszczalnego, gdyż każdy z punktów jest dopuszczalny. Sprawdzono działanie algorytmu hybry-

Table 2. Wydajność i skuteczność różnych algorytmów optymalizacji w projektowaniu falowodu.

Algorytm	Powella	CRS2+ Powell ¹	CRS2+ Powell²	CRS2
<i>n</i> : liczba oszacowań ³ $f(\cdot)$	167	223	338	1000
<i>r</i> : skuteczność ⁴	8%	38%	58%	49%
<i>N</i> : nakład obliczeniowy ⁵	6000	1397	1167	4449

¹Przełączenie po 130 oszacowaniach $f(\cdot)$ przez CRS2

²Przełączenie po 260 oszacowaniach $f(\cdot)$ przez CRS2

³Wartość średnia, dla pojedynczego procesu optymalizacji

⁴Odsetek pomyślnych optymalizacji (tj. $f(\mathbf{x}^*) < 0,012$)

⁵Nakład obliczeniowy (liczba oszacowań $f(\cdot)$ potrzebnych

do uzyskania pomyślnego wyniku z *p*-stwem większym od 0,95)

dowego, gdy przełączenie z CRS2 na Powella następuje po 130 albo 260 oszacowaniach $f(\cdot)$ dokonanych przez CRS2. Porównanie skuteczności i wydajności algorytmu hybrydowego (oraz metod składowych działających samotnie) zamieszczono w tabeli 2. Istotnym wnioskiem jest to, że opłaca się stosować metodę hybry-

dową (dowolna wersja działa lepiej niż każdy z algorytmów składowych w pojedynkę), oraz że wybór chwili przełączenia ma znaczenie.

Należy tutaj przyznać, że powyższa relacja dotyczy ostatecznego doboru algorytmu dla pełnego modelu falowodu. Natomiast wnioski dotyczące konstrukcji algorytmu hybrydowego uzyskane wcześniej, dla modelu uproszczonego, okazały się zupełnie niesłuszne dla modelu pełnego (wnioskowano m.in. używanie wyłącznie algorytmu CRS2). To wskazuje, jak niezwykle istotne jest, aby model uproszczony nie zatajał istotnych cech modelu pełnego.

5. PODSUMOWANIE

Należy zaznaczyć, iż w opinii ekspertów z dziedzin właściwych modelom, przedstawione tutaj problemy optymalizacji uchodziły za bardzo trudne do rozwiązania, szczególnie przy niezbyt wysokim nakładzie obliczeniowym. Zastosowanie metodyki autora rzeczywiście doprowadziło do ich rozwiązania, co samo w sobie jest sukcesem. Efektywność końcowych algorytmów hybrydowych jest podobnego rzędu, co komercyjnych pakietów optymalizacji (przy czym nie zawsze możliwe było zastosowanie wyłącznie ilościowych kryteriów porównania), a ponadto mają tę zaletę, że pozostają pod pełną kontrolą ich twórcy. Jest to szczególnie istotne, jeśli przewiduje się dalsze prace badawcze na ten temat.

Opracowana procedura projektowa może również okazać się interesująca w obliczu problemu optymalizacyjnego dla specjalistów z różnych dziedzin, którzy nie posiadają gruntownej wiedzy z zakresu optymalizacji, Wystarczy nieco praktyki w programowaniu aby samodzielnie wykonać solwer specjalizowany do interesującego nas typu zadań.

Dalszy kierunek ewentualnych prac wyznacza pokusa, by zaprezentowaną procedurę projektową zakodować w komputerze tak, by właściwy algorytm hybrydowy sam tworzył się i wykonywał napotkawszy konkretny problem. Jest to zadanie ambitne. Oprócz nietrywialnego algorytmu selekcji właściwych metod składowych (zastępującego de facto inteligencję i intuicję inżyniera), wiele wysiłku musi być włożone w standaryzację i wypromowanie jednolitej formy zapisu wszystkich istotnych cech zadania i solwera. Tylko wówczas bowiem można zaprojektować algorytm automatycznie wybierający metodę najlepiej dostosowaną do problemu. Wybór taki jest wstępny, ostateczną weryfikacją przydatności metody jest skuteczność (którą również należałoby mierzyć w jakiś standardowy sposób). Dlatego też już podczas działania algorytmu wstępnie wybranego należy przeznaczać stałą, niewielką część nakładu obliczeniowego na testowanie metod teoretycznie nie najlepiej dopasowanych — i przełączać się na nie gdy tylko okazują się lepsze. Wstępne testy prototypu takiego środowiska

obliczeniowego potwierdzają słuszność postawionych tu postulatów.

6. BIBLIOGRAFIA

DESIGN OPTIMISATION UNDER COMPUTABILITY CONSTRAINTS

Abstract: Physical phenomena that determine behaviour of systems often have to be modelled not analytically but by solving some problem numerically. Such modelling is necessary if one is going to design a new object or find settings for the existing one so that its performance is maximised. Many undesired features exhibited by numerical models make it difficult to employ an optimisation algorithm seeking optimal performance. Those features are: simulation noise, switching and model output computation failures, termed in this paper computability constraints. A design procedure is proposed here that makes possible creation of an effective hybrid optimisation routine tailored to a specific design optimisation task. Results of application of such a universal routine to two practical cases are studied. Hints are given for further automation of the proposed design procedure.

References

- [1] Synaps, Inc. Epsy 2003. User's Guide. Atlanta.
- [2] Findeisen W., Bailey F.N., Brdyś M., Malinowski K., Tatjewski P., Woźniak A. (1980) Control and Coordination in Hierarchical Systems. John Wiley & Sons, Inc.
- [3] Kamola M. (2004) Algorithms for Optimisation Problems with Implicit and Feasibility Constraints. Rozprawa doktorska, Wydział Elektroniki i Technik Informacyjnych, Politechnika Warszawska.
- [4] Papalambros P.Y. (1988) Principles of optimal design. Cambridge University Press.
- [5] Parkinson A., Balling R., Free J., Talbert J., Davidson D., Gritton G., Borup L., Busaker B. (1992) OptdesXTM; A Software System for Optimal Engineering Design. Users Manual. Release 1.0. Design Synthesis, Inc., Provo.
- [6] QWED, Ltd. (2003) QuickWave-3D User's Manual. Warszawa.
- [7] Radwański E., Skowroński R., Twarowski A. (1993) Problemy modelowania systemów energotechnologicznych. Instytut Techniki Ciepłej, Politechnika Warszawska, Warszawa.