15. SYMPOSIUM „DESIGN FOR X“
NEUKIRCHEN, 14. UND 15. OKTOBER 2004

PRODUKTSTRUKTURBASIERTE PRODUKTENTWICKLUNGSPROZESSMODELLIERUNG UND OPTIMIERUNG

Tamás Rick, Márk Horváth, Tibor Bercsey

Zusammenfassung

Im 21. Jahrhundert können wegen der immer steigenden und wechselnden Kundenwünsche nur solche Firmen wettbewerbsfähig bleiben, die am schnellsten die Kundenanforderungen befriedigen können. Die dazu notwendige Basis ist eine grobe Produktstruktur aus der Konzeptphase mit vorgesehenen Kosten, Zeiten und Termine.

Zur Produktstrukturbeschreibung wird die Design Structure Matrix (DSM) verwendet, sie basiert auf der Annahme, dass in Entwicklung und Konstruktion größere Rationalisierungspotentiale im Zusammenspiel und der Reihenfolge der Engineering-Tätigkeiten (Entwicklung eine Teil der Produktes) als in der effizienteren Durchführung einzelner Konstruktions-(Teil)Aufgaben liegen.

Im günstigen Fall können durch Vertauschen von Reihenfolgepositionen Iterationen vermeiden oder verkürzt werden. Diese Reihenfolge-Optimierung erfolgt mit Hilfe genetischer Algorithmen. Als Selektionsverfahren werden „Better Half“ und „Tournament“ Selektion für Rekombination ein Position basierte und für Mutation ein Reihenfolge basiert Verfahren angewendet. Die Auswertung der Ergebnisse erfolgt durch einfache Durchlaufzeitberechnung, damit nützliche Informationen für Ressourcenplanung und Netzpläne erhalten werden.

Diese Forschungsarbeit wurde in Rahmen der T03247 OTKA durchgeführt.

1 Einleitung

Der Erfolg eines Unternehmens wird durch seine Fähigkeit bestimmt, sich veränderten Marktanforderungen anpassen zu können. Diese Fähigkeit kommt sowohl durch Qualität und Preis des Produktes, als auch durch die Geschwindigkeit, mit der ein Unternehmen auf Marktbedürfnisse reagieren kann, zum Ausdruck. Marktkonforme Produkte und kurze Entwicklungszeiten, niedrige Entwicklungskosten sind somit wesentliche Wettbewerbsfaktoren. Für die Stärkung der Wettbewerbsfähigkeiten kommt deshalb dem Produktentwicklungsprozess als dem Prozess, in dessen Rahmen über Produkteigenschaften und Entwicklungszeiten entschieden wird, eine besondere Bedeutung zu. Der Prozess kennzeichnet das eigentliche Vorgehen im Projekt zur Herstellung des Produkts; er beschreibt also den Planungs- und Realisierungsablauf. Im Prozess werden die für die Zielerreichung notwendigen Aktivitäten – gemeinhin als Arbeitspakete bezeichnet – in definierte Abläufe eingeordnet, wobei die jeweils notwendigen Vorgaben sowie die zu erreichenden Ergebnisse bindend festgelegt sind. Die Abläufe und Konstruktionsarten hängen zusammen.

Wenn die Bearbeitungstiefe eines Produkts unterschiedlich groß ist, kann man verschiedene Konstruktionsarten (Neu-/Anpassungs-/Variantenkonstruktion) definieren. Diese sind dadurch gekennzeichnet, dass die Konstruktionsphasen unterschiedlich intensiv und teilweise gar nicht durchlaufen werden. Man bezeichnet eine Konstruktionsaufgabe als eine Neukonstruktion, wenn alle drei Phasen: Konzipieren, Entwerfen und Ausarbeiten in gleicher Weise neu zu bearbeiten sind. Eine VDMA-Befragung hat ergeben, dass nur ca. 10% der Konstruktionsaufgaben der Neukonstruktion zuzureichen sind, ca. 15% der Anpassungskonstruktion und ca. 70% der Variantenkonstruktion [1].

Aus den Zahlen ist es klar, dass es eine sehr wichtige Aufgabe ist, Hilfsmittel für solche Konstruktionsaufgaben wie Anpassungs- und Variantenkonstruktion herzustellen. Diese Hilfsmittel sollen eine dynamische, flexible Prozessplanung ermöglichen für Management (Ressourcen-, Termin- und Kostenplanung) und für Konstrukteure. Sie sollen eine optimale Planung von den Prozessen gewährleisten, damit die Termine und Kosten bei veränderten Marktanforderungen und Kundenwünschen eingehalten werden können [2].

Das Endergebnis eines Entwicklungsprozesses ist ein Produkt. Der soll den Aufbau des Prozesses bestimmen, deswegen soll als Basis für Prozessplanung die Produktstruktur, bzw. Produktelemente-Struktur dienen [2]. Für die Beschreibung der Strukturelemente und deren Zusammenhänge ist die Design Structure Matrix ein innovatives Werkzeug. Großer Vorteil der DSM ist die Möglichkeit, mehrere Informationen für weitere Arbeiten (z.B.: Finanzierungsplanung, Terminplanung, Ressourcenplanung, EDM/PDM Systemen) abzuleiten.

2 Design Structure Matrix (DSM)

Stewart [3] war der erste, der zur Erfassung der Ablauforganisationen und Informationsflüsse DSM benutzte. Später hat dazu Rogers [4] ein auf genetischen Algorithmen basierendes Optimierungssystem entwickelt, um Entwicklungs- und Konstruktionsprozesse zu optimieren. Die DSM Methode basiert auf der Annahme, dass in Entwicklung und Konstruktion größere Rationalisierungspotentiale im Zusammenspiel und der Reihenfolge der Engineering-Tätigkeiten (Entwicklung eines Teils des Produktes) in einer effizienteren Durchführung einzelner Konstruktions-(Teil)Aufgaben liegen.

Die für die Entwicklung und Konstruktion typischen iterativen Arbeitsläufe mit ihren Rückkopplungen und Optimierungsschleifen werden dafür zunächst in einer angedeuteten Informationsflussmatrix abgebildet, deren Zeilen- und Spaltenköpfe die Produktstruktur-Elemente in ihrer gegenwärtigen Ablauffolge darstellen. Die Entwicklung von Strukturelementen benötigt Input-Informationen von den in der jeweiligen Zeile mit einem „1“ markierten Funktionen und leitet in gleicher Weise Output-Informationen an andere Strukturelemente weiter. In der Matrix, die Elementen die über den Diagonal sind, sind Vorwärtskopplungen, z.B.: von Strukturelement (SE.) 1. wird Information zum SE. 2. und 6. weitergeleitet. Unter der Diagonale sind Rückkopplungen oder Zyklen, aber diese Angaben sind nur für die Reihenfolge der Strukturelemente. Im günstigen Fall können durch Vertauschen von Reihenfolgepositionen Iterationen vermeiden oder verkürzt werden. Für diese Aufgabe wird ein genetischer Algorithmus implementiert.

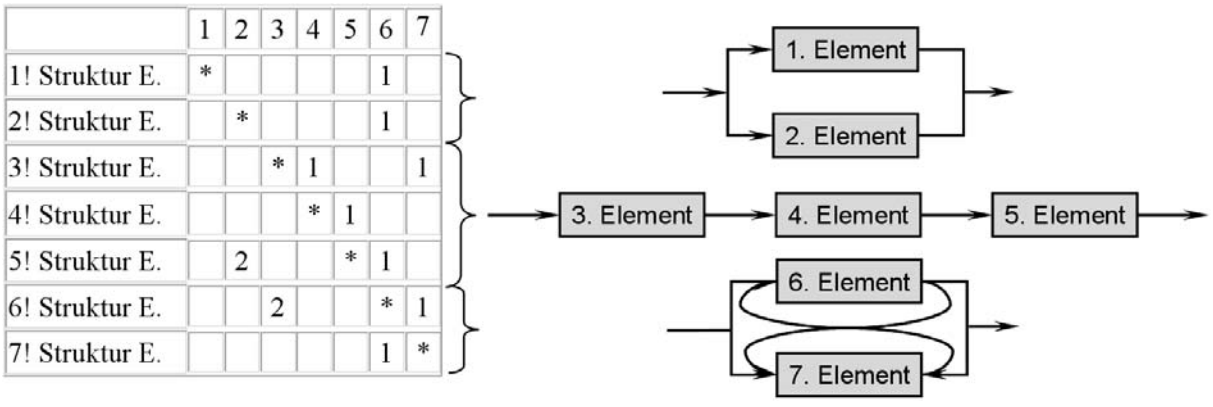

Bild 1: Design Structure Matrix

Ein weiteres Problem bei den Prozessen bzw. Informationsflüssen ist, wenn diese sich schneiden (Crossover/Kreuzung). Dass heißt, das die Zyklen ineinander gekoppelt sind, im Bild 1, unter der Diagonale von 6 zu 3 (2) und von 5 zu 2 (2) sind solche ineinander gekop-

pelte Zyklen. Diese können zum Chaos durch nicht klare Informationsflüsse im Prozess führen, damit wird die Prozesszeit erheblich länger und natürlich werden die Prozesskosten auch die vorgesehenen Werte überschreiten.

3 Genetische Algorithmen

In den frühen 60er Jahren versuchen John Holland und Ingo Rechenberg unabhängig von einander, den Prozess der natürlichen Evolution in einem Algorithmus zu programmieren. Rechenbergs Ansatz wurde unter dem Namen „Evolutionsstrategie“ [5] bekannt, der von Holland unter dem Namen „Genetic Algorithm“[6]. Genetischen Algorithmen sind sehr gute Werkzeuge für diskrete Optimierung. Genetische Algorithmen arbeiten nicht immer direkt mit den zu optimierenden Parametern, sondern mit der binär kodierten Form dieser Parameter, die für numerische Probleme zwar gut anwendbar aber für Reihenfolgeoptimierung nicht geeignet sind [7], [8].

Somit wird ein Produkt durch eine endliche Zeichenkette, die dem natürlichen Chromosom entspricht, abgebildet. In Anlehnung an die Natur wird diese Zeichenkette Chromosom genannt. Jeder Punkt des Lösungsraumes, der alle Produktvarianten beinhaltet, kann durch ein Chromosom dargestellt werden. Die Suche des Genetischen Algorithmus erfolgt von einer zufällig erzeugten Startpopulation, nicht von einem einzelnen Lösungspunkt aus.

3.1 Selektion

Die Individuen der Startpopulation werden bewertet und bekommen einen Fitnesswert zugeordnet. Der Fitnesswert eines jeden Individuums ist in unserem Fall die Prozessdurchlaufzeit. Danach werden die Individuen zur Reproduktion zufallsgesteuert selektiert, wobei Individuen mit einem höheren Fitnesswert mit einer größeren Wahrscheinlichkeit ausgewählt werden. In unserem Beispiel haben wir zwei verschiedene Selektionsverfahren ausprobiert:

- „Better Half“ Selektion: die einzelnen Individuen werden ausgewertet und nur die beste Hälfte der Population (Fitnesswert) wird zur Rekombination ausgewählt [9].

- „Tournament“ Selektion: die einzelnen Individuen werden ausgewertet, und es werden immer nur zwei ausgewählt und deren Fitnesswert wird verglichen, und der beste wird für Rekombination ausgewählt [10].

3.2 Rekombination

Die ausgewählten Individuen bilden die so genannte Elterngeneration, die durch Rekombination infolge „Crossing over“ Nachkommen erzeugt. Bei der Rekombination werden lediglich genetische Informationen zwischen den Chromosomen ausgetauscht. In unserem Fall ist die Rekombination eine positionsbasierte Rekombination [11]. Einige Positionen werden von dem ersten Elternindividuum zufällig ausgewählt und in dem Kind an dieselbe Position geschrieben. Danach werden die Lücken mit der Nummern der Strukturelemente aus dem zweiten Elternindividuum ausgefüllt, Bild 2.

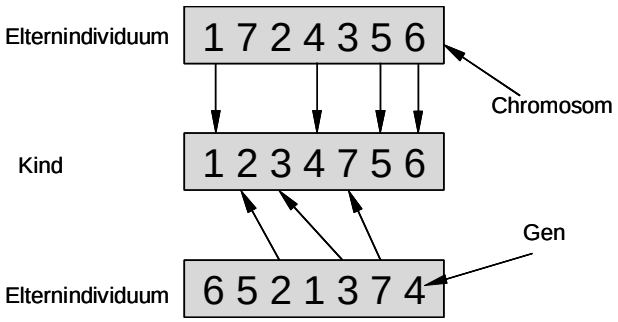

Bild 2: Ablauf der Rekombination

Nach der Rekombination und Mutation der Elemente der Zeichenkette werden ihre ursprünglichen Verknüpfungen mit anderen Strukturelementen beibehalten. Nur diejenige Zyklenzahlen werden auf „1“ gesetzt die über dem Diagonal kommen.

3.3 Mutation

Mutation erfolgt reihenfolge-basiert [11]. Eine Position wird zufällig für die Mutation ausgewählt und mit einer – auch zufällig – ausgewählten Position ausgetauscht, Bild 3.

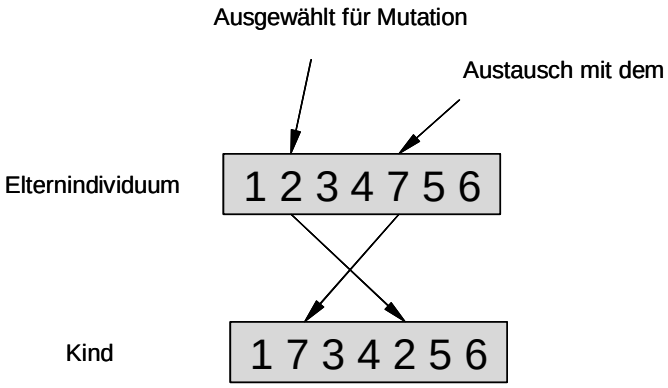

Bild 3: Ablauf der Mutation

3.4 Auswertung

Als Auswertung (Fitnesswertbestimmung) wird die Prozessdurchlaufzeit mit Zyklen berechnet.

In unserem Beispiel haben wir eine Lernrate festgelegt (0,95), die zu einer besseren Annäherung an den realen Prozess führt. Das bedeutet, dass beim Berechnen der Zyklen die Zyklendurchlaufzeiten immer kürzer werden. Diese Berechnung erfolgt nach der bekannten geometrischen Reihe:

Formel:

$$T_{\Sigma} = \sum_{i=1}^{n} t_i \frac{1 - L_{rate}^{n}}{1 - L_{rate}} \tag{1}$$

Wo:

- T_Σ = Zyklusdurchlaufzeit [Zeiteinheit]
- t_i = Arbeitszeit für ein Strukturelement [Zeiteinheit]
- L_{rate} = Lernrate

Diese Berechnung soll für alle Zyklen wiederholt werden. Bei übergeordneten Zyklen werden die Subzyklen summiert.

Die nun erzeugte neue Population durchläuft den iterativen Prozess Selektion – Reproduktion – Rekombination – Mutation – Bewertung solange, bis das Abbruchkriterium erfüllt ist. Als Abbruchkriterien können entweder die maximale Anzahl der Generationen oder der zu erreichende Fitnesswert, der dem Zielfunktionswert entspricht, aufgestellt werden.

3.5 Parameter des Algorithmus

Es wird allgemein behauptet, dass die Funktionsparameter der genetischen Algorithmen aufgabeabhängig sind.

Für unsere Aufgabe waren die folgenden Werte am besten geeignet:

- Populationsgröße: 20
- Generationszahl: 1000
- Wahrscheinlichkeiten:
  - Mutation: 0,2
  - Rekombination: 0,95

Bei diesen Parametereinstellungen hat ein Optimierungsgang nicht mehr als 3-4 Minuten gedauert.

4 Ergebnisse

Als Beispiel wurde eine DSM mit 22 Elementen erstellt, wo die Zyklenzahlen und Arbeitszeiten (T) für die einzelnen Elemente angegeben wurden. Das Ziel der Optimierung war, eine verkürzte Prozessdurchlaufzeit zu erreichen bei der minimalen Anzahl der Rückkopplungen und Kreuzungen.

Die Ausgangs-DSM hatte 24 Rückkopplungen und 16 Kreuzungen, bei einer Durchlaufzeit von 12792 [Zeiteinheiten], siehe Bild 4. Die optimierte DSM hat nur 10 Rückkopplungen und 2 Kreuzungen bei einer Durchlaufzeit von 3767 [Zeiteinheiten], siehe Bild 5.

P.E.	T	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	30									1													
2	40	8																					
3	10																		1				
4	10														1								
5	10								1														
6	50	4		2					1												1	1	
7	40														1								
8	50		8					6		1													
9	20				7				2														
10	20								8						1								
11	40						8																
12	30											5											
13	30												3										1
14	20													6									
15	30								4						8								
16	20															6		1					
17	30										8						7		1				
18	40					6												8		1			
19	50																		6		1		
20	40								7						4					2			
21	20																8						
22	20																						

Bild 4: Ausgangs-DSM

P.E.	T	12	11	6	3	21	20	19	18	17	16	10	15	5	13	14	9	4	8	7	2	1	22
12	30		1																				
11	40			1																			
6	50				1	1	1												1			1	
3	10								1														
21	20										1												
20	40							1								1			1				
19	50						1		1														
18	40							1		1				1									
17	30								1		1	1											
16	20									1			1										
10	20															1			1				
15	30															1			1				
5	10																		1				
13	30	3																					1
14	20														6								
9	20																						
4	10															1							
8	50																1			1	1		
7	40															1							
2	40																					1	
1	30																1						
22	20																						

Bild 5: Optimierte DSM

Die Zahl der Rückkopplungen und Kreuzungen wurden nicht in der Zielfunktion berücksichtigt, und nur die Rückkopplungen sind in der Durchlaufzeitberechnung integriert. Zurzeit unsere Implementierung die Kreuzungen und die Kosten nicht behandelt.

Der Algorithmus ohne die Anwendung eines getrennten Clustering Algorithmus hat zusammenhänge zwischen den Strukturelementen erkannt und diese zu Subprozessen zusammengefasst (z.B.: 19-18-17-16 Elemente).

Die zwei verschiedenen Selektionsverfahren (Bild 6 und Bild 7) haben in den Optimierungsverfahren die gleichen Ergebnisse ergeben, aber wir können nicht eindeutig sagen, welche Selektion für diese Aufgabe besser geeignet ist. Um diese Aussage zu treffen, möchten wir in zukünftigen Optimierungen beide einsetzen, damit späterhin der Effizienter einsetzen zu können.

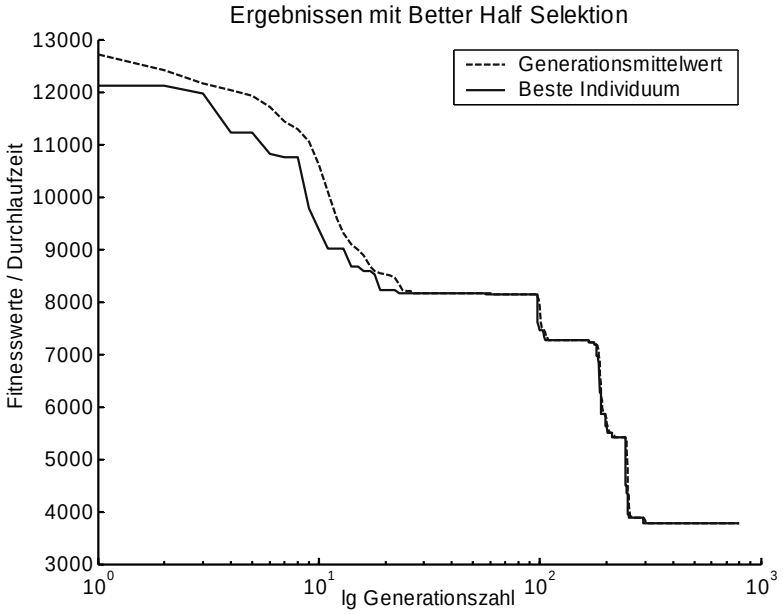

Bild 6: Optimierungsabläufe mit Better Half Selektion

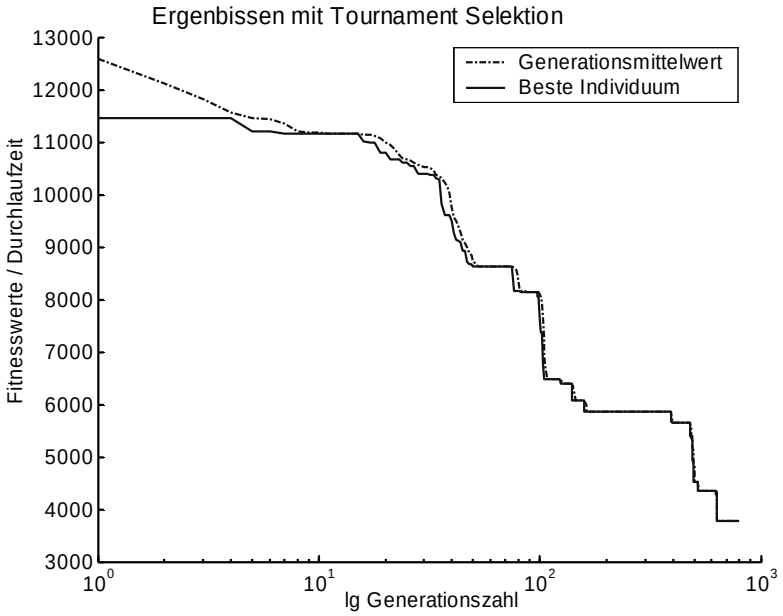

Bild 7: Optimierungsabläufe mit Tournament Selektion

Nach der Durchführung von Testläufen hat sich gezeigt, dass das von uns entwickelte Optimierungssystem das Reihenfolge-Optimierungsproblem mit Erfolg gelöst hat. Die Endergebnisse sind ca. 75% besser als die Ausgangsreihenfolgen. Diese Ergebnisse zeigen, dass das DSM mit genetischen Algorithmen ein gutes Hilfsmittel für Produktentwicklungsprozessplanung und Optimierung ist.

Obwohl die genetischen Algorithmen robust sind, sind sie schnell und erzeugen viele mit annehmbaren Fehlern belastete Lösungen. Ferner wird das System für Multi Objekt Optimierungssystem (Zeiten, Kosten und Ressourcenplanung) weiterentwickelt, damit wird ein ganzheitliches Planungs- und Optimierungssystem für Prozesse verwirklicht.

5 Literatur

[1] Ehrlenspiel, K.: Integrierte Produktentwicklung: Methoden für Prozessorganisation, Produktentwicklung und Konstruktion, Carls Hansen Verlag, München, Wien, 1995

[2] Reinertsen, D.G.: Die neuen Werkzeuge der Produktentwicklung. Carl Hanser Verlag, München, Wien, 1998

[3] Steward, D.V.: SystemAnalysis and Management: Structure, Strategy and Design. Petrocelli Books Inc. 1981

[4] Rogers, J.L.: DeMAID/GA – An Enhanced Design Manager's Aid for Intelligent Decomposition, 6th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Seattle, WA, September 4-6, 1996a. AIAA paper No. 96-4157, 1996

[5] Rechenberg, I.: Evolutionsstrategie – Optimierung technischer Systeme nach Prinzipen der biologischen Evolution. Friedrich Frommann Verlag, Stuttgart, 1973

[6] Holland, J.: Adaptation in natural and artificial systems. MIT Press, Cambridge, Mass 1975

[7] Altus, S.S.; Kroo, I.M.; Gage, P.J.: A Genetic Algorithm for Scheduling and Decomposition of Multidisciplinary Design Problems. ASME Paper 95-141., 1995

[8] Bloebaum, C.L.: An Intelligent Decomposition Approach for Coupled Engineering Systems. In: Proceedings of the Fourth AIAA/AF/NASA/OAI Symposium on Multidisciplinary Analysis and Optimization, Cleveland, OH, 1992

[9] Abramson, D., Lewis, A., Peachey, T., Fletcher, C.: An Automatic Design Optimization Tool and its Application to Computational Fluid Dynamics. Proc., ACM/IEEE SC2001 Conf., Denver, CO, 2001

[10] Back, T.: Generalized Convergence Models for Tournament- and Selection. In: Larry J. Eshelman (ed.), Proc. Sixth Int. Conf. on Genetic Algorithms, 2-8. Morgan Kaufmann Publishers, San Francisco, 1995

[11] Syswerda, G.: Schedule Optimization Using Genetic Algorithms, Handbook of Genetic Algorithms, Van Nostran Reinhold, New York, 1990

Prof. Dr.-Ing. Tibor Bercsey
Dipl.-Ing. Tamás Rick
Lehrstuhl für Maschinenkonstruktionslehre
Technische und Wirtschaftwissenschaftliche Universität Budapest
Bertalan L. Str. 3, H-1111Budapest
Tel: +36-1-463-1372
Fax: +36-1-463-3505
Email: bercsey.tibor@gszi.bme.hu
rick.tamas@gszi.bme.hu
URL: http://www.gszi.bme.hu

cand. Dipl.-Math. Márk Horváth
Faculty of Sciences
Vrije Universität Amsterdam
De Boelelaan 1081a
1081 HV Amsterdam
Email: cyber@inf.elte.hu
URL: http://people.inf.elte.hu/cyber/