

## ZAJÍMAVOSTI ZE SVĚTA VĚDECKOTECHNICKÝCH VÝPOČTŮ

### 1. Úvod.

#### 1.1 Definice pojmů:

Lidé něco objeví, vynalezou, zhotoví a provozují. Po nějaké době provozu odborníci usoudí, že vynález neodporuje v současné době platným přírodním zákonům; jakmile pak ještě příslušní úředníci soujmou k vynálezu postoj zcela blahovolný, chopí se věci škola a upraví ji pro své potřeby. Především bývá nutno z hlediska výuky (a hlavně zkoušení) všechno řádně utřídit, systemizovat a definovat. Tak tomu bylo i s výpočetní technikou.

Vysvětluje se tedy dychtivým studentům, že na počítači lze provádět

- a) vědeckotechnické výpočty (VTV)
- b) ekonomickomatematické výpočty (EMV)
- c) zpracování hromadných dat (ZHD).

Původně se tyto třídy prací charakterizovaly objemem vstupů a výstupů (ad "a" nejmenší, ad "c" největší) a složitostí algoritmů (ad "a" nejsložitější, ad "c" jednoduché). Praxe však nebraha na školu ohled a tak dnes máme VTV s rozsáhlými vstupy a výstupy i ZHD s velice složitými algoritmy. Na teoretické frontě se svádějí tuhá klání a praxe si vesele vede po svém.

Tato situace značně ztěžuje psaní úvodu k tomuto příspěvku. Autor je nucen požádat čtenáře o jistou shovívavost; pod pojmem VTV si budeme v dalších úvahách představovat všechno, co tam podle našeho programátorského cítění patří (včetně příslušného dílu ZHV). Objem vstupů a výstupů nebude důležitý, spíše nám půjde o výpočty s matematicky i programátorsky složitými algoritmy.

## 1.2 Automatizované systémy řízení a VTV:

V poslední době nás vývoj přivádí do zajímavé situace: VTV pronikají do úloh ZHD. Zvláště se to projevuje při tvorbě automatizovaných systémů řízení (ASŘ). Ponecháme stranou rozbor pojmu ASŘ jakožto zaklínadla nebo módního slova, užívaného ponejvíce v písemných i ústních projevech anotářů a samouků.

Poznámka: bylo by velmi zajímavé zpracovat sociologickou studii o módních "odborných" výzvech a jejich vývoji; jen namátkou uvedme z oblíbených pojmů posledních let časovou řadu "síťová analýza - systémové inženýrství - databanky - ASŘ", jejíž jednotlivé prvky vědy sloužily k řešení čehokoliv a k posvědnutí téhož na novou, kvalitativně vyšší úroveň. Úvahy na toto téma se však svým rozsahem vymykají rámci tohoto stručného příspěvku.

Dále tedy předpokládáme reálnou existenci ASŘ jakožto systému, který je třeba řešit (nikoliv jako řečnického obratu). Kdo se problémem blíže zabývá sjišťuje, že výpočetní systémy budou v ASŘ provádět řadu úloh, které mají charakter typických VTV. Tyto úlohy budou s úlohami ZHD přímo propojeny a budou pracovat na společných souborech dat.

## 1.3 Poznámka k uplatnění VTV:

V automatizovaných systémech řízení se uplatní VTV nejrozmanitějšího druhu, od obecných úloh na př. statistiky, optimalizace a pod. až po specializované výpočty příslušného oboru (na př. výpočet parametrů družicových spojů - viz kapitola 6 tohoto příspěvku).

Nové pole se zde otevírá úlohám z oboru operační analýzy. Jak je všeobecně známo, byl operační výzkum považován svého času za vrchol všeho a za jediný možný nástroj řízení. Bylo vyplýváno mnoho času na praktické aplikace různých dopravních problémů, teorií front a her a dalších důmyslných hříček. Očekávané výsledky se v praxi nedostavily. Důvody neúspěchů byly shruba tři:

- umíněnost teoretiků, kteří odmítali přizpůsobit řešení praxi
  - nedostatek správných a platných dat na vstupu úloh (takže se pracovalo velice přesně s vymyšlenými nebo v nejlepším případě hrubě odhadnutými daty)
  - organizační a provozní nepružnost: od provozních pracovníků se vyšadovala práce navíc (vyplňování formulářů, shánění roztodivných údajů); zpracování bylo zajišťováno většinou v odlehlých střediscích a odezva systému byla příliš dlouhá.
- Zapojení uvedených metod do ASŘ neodstraníme sice první zá vadu, avšak vzpurné teoretiky lze úspěšně ukásnit metodou cukru a biče. Rychlý čtenář jistě již sám odhalil, že zcela zmizí problémy s daty i organizační nepružnost. Úlohy operačního výzkumu tedy nacházejí nové uplatnění; věříme, že právě v ASŘ prokáží svou užitečnost.

#### 1.4 Závěr úvodu:

Tento příspěvek si klade za cíl poněkud oprášíit problematiku VTV a podívat se na ni z hlediska nových aplikací.

## 2. Technika se vyvíjí.

Původní obrovská počítačící monstra (pro která neměla čeština ještě název) byla určena pro vědeckotechnické výpočty. Nějaký čas se nevědělo, co s tím vším dále; pak se najednou vývoj doslova splašil. Stroje se zmanžovaly přímo před očima, učily se a rozum braly. Začaly být využívány ke všemu možnému a oblast VTV byla vytlačována k okraji zájmu.

Zároveň se odehrávala terminologická čili názvoslovná tragédie. Ponechme bez komentáře bezuzdné řádění novinářů, jejichž "komputery", "samočinné počítačící stroje", "elektronkové mozky"

a podobné urážky jazyka i techniky nás budou ještě dlouho děsit. Když se odborníci konečně shodli na názvu "počítač", byly již na trhu výrobky, které se takto označit mohly, ale také nemusely. Sotva bylo ustaveno rozdělení "velké - střední - malé", začaly se prodávat počítače kapesní a stolní. Zoufalí češtináři nechali do názvosloví proniknout předponu "mini-"; mezitím se začaly počítače zabudovávat do náramkových hodinek. Krásný český pojem "mikroprocesor" slaví úspěchy (ačkoliv málokdo pořádně ví, co to je). Průlom cizích slov se nepodařilo zastavit a to vedlo k dalšímu dějství dramatu: ve světě si jednotlivé firmy tvoří vlastní názvosloví a protože je naše země proslulá sbírkou výrobků nejrůznějších firem, stává se i názvoslovnou džunglí. Nedivme se pak, že v odborném tisku nalezneme vážné míněné polemiky o významu pojmu "item", "set" a podobně.

V praxi má toto smatení jazyků velmi nepříjemné důsledky; zejména odborná literatura našeho oboru je tímto stavem více než poznamenána (vezměme jako ukázkou slovenskou edici "Počítače", kde velmi hodnotná díla špičkových světových autorů byla často nevhodným překladem dovedena až do nerozumitelnosti; otrlým jedincům doporučujeme zevrubné studium časopisu MAA, který ponechává v názvoslovné oblasti volné pole působnosti autorům, překladatelům i těm, kteří se za ně považují).

Prudký vývoj techniky nemá však důsledky jen v terminologickém zmatku. Praxe musela přehodnotit i řadu tradičních přístupů a pojetí. Jestliže u prvních typů počítačů byl běžný režim "open-shop" (hle, další jazyková perla!), pak u novějších strojů musíme vzhledem k ekonomice provozu zavést tvrdý uzavřený režim továrny na data (ať se to účastníkům různých seminářů líbí nebo ne). Vtom nám ale vpadly do zad stolní a kapesní počítače, které jsou nasmírně chytré a pojem "close" neznají.

Právě v oblasti VTV natropily tyto maličké chytré bestie mnoho rozruchu. Vědci a technici se naučili programovat, ochpčili si (alespoň se to domnívali) tyto moudré stroječky a jali se vysmívat profesionálním programátorům. K této situaci se máním vrátit v kapitole 4.

Ale už tu máme další novinku. Hardwaroví kouzelníci vytáhli z klobouku inteligentní terminály, které dávají uživateli pocit vlastnictví celé kapacity velkého stroje.

VTV teď můžeme zpracovávat

- zadáním do výpočetního střediska (v dávkovém režimu)
- na stolním nebo kapsním počítači
- na větším stroji on-line v režimu real-time pomocí obyčejného nebo inteligentního terminálu.

(Poslední větu doporučuji pozorností brusičů jazyka).

Z praktických zkušeností mohu doporučit využívání všech uvedených možností ve vhodné kombinaci.

Zde upozorňuji na nebezpečí podcenění některých prostředků. Na VŠB v Praze byla zpracována zajímavá diplomová práce na téma "Využití kapsních kalkulaček v ASŘ" (lit./6/), která na význam a možnosti některých výkřiků techniky upozorňuje a snaží se najít jejich seriózní uplatnění.

### 3. Programátorské starosti.

#### 3.1 Všeobecně:

Představme si, že jsme programátory VTV a zjišťujeme hlavní problémy, na které narazíme při aplikaci VTV v automatizovaném systému řízení. Nebude jich málo a tak se v dalším textu omezíme jen na ty zásadní. Moudrý programátor hromadných dat nám bude nahlížet přes rameno a možná, že si vybere leccos užitečného i pro svou práci.

#### 3.2 Programovací prostředky:

Věřím, že zanícení vyznavači Assembleru již dávno přešli k systémovým programátorům nebo se stali poustevníky. My realisté máme pro VTV k dispozici Fortran, Basic a PL/1. Víme, že APL je krásná věc, ale není většinou k máni; vyhne se i specialistům, kteří pracují s jazyky typu Simula a budeme se věnovat obecným pravdám.

Basic je velice lákavý. Naučí se v něm programovat s promínutím každý; pro rutinovaného programátora bývá rajským snem ladění programů v Basicu. Pozor však na to, že většina imple-

mentací funguje jako interpret a vlastní výpočet je 10 až 20krát pomalejší než práce podle téhož algoritmu, zapsaného ve Fortranu.

Solidnímu programátoru VTV stejně nezbyvá, než se pořádně naučit Fortran. V praxi nalezneme nejrozličnější verze tohoto 26 let starého jazyka: od poměrně elementární implementace na EC 1010 až po vyšperkovaný "Fortran IV plus" na SMEP 3 a 4.

Zastánci PL/1 bývají nekompromisní a velice popudliví. Snad je to tím, že se museli naučit tak složitý jazyk. Prohlašují, že PL/1 je jazyk vynikající (máme-li k dispozici vynikající kompilátor), oplývající finesami; jestliže někdo tento jazyk opravdu ovládne (v tom bývá potíž), nebude mít ve světě VTV konkurenci. Věřím, že tímto tvrzením jsem složil povinný hold vyznavačům PL/1 a tak preventivně sabránil jejich útokům.

### 3.3 Datové základny:

Novější aplikace VTV vyžadují často poměrně velké soubory dat. Při aplikacích v ASŘ pracujeme dokonce s hotovými soubory, vzniklými v jiných úlohách. Vědeckotechnický programátor je nucen sestoupit a výšle složitě matematiky a algoritmických propletenců a věnovat se studiu datových struktur a práce s nimi. Pro útěchu doporučuji zavilým vědcům lit./9/; získají hrdé vědomí, že soubory dat jsou teoreticky velice náročná disciplína a práce s nimi může zamotat hlavu i vysoce fundovanému pracovníkovi.

Programátoři VTV řeší i některé problémy, neznámé v oblasti hromadných dat. Jako příklad si uveďme ukládání rozsáhlých matic na paměťová média (po řádcích, po sloupcích nebo jinak?). Pokud jde o časově náročné výpočty, bývá zde účelné kalkulovat s časy přístupů k datům a vhodnou strukturou souboru omezit tyto časy na minimum.

Upozorňuji též na užitečný trik, zvaný "virtuální pole" (nebo i jinak). Pole (ARRAY), které se nevejde do vnitřní paměti rozdělíme na části a uložíme na disk. Přístup k prvku pole provádí na př. fortranská FUNCTION, jejímiž parametry jsou hodnoty všech indexů prvku. Rutina současně eviduje,

kteřá část pole je právě ve vnitřní paměti a pracuje takto:

a. podle hodnot indexů zjistí, v které části pole žádaný prvek leží;

b. je-li tato část pole ve vnitřní paměti, přejdi k "c"; není-li, přečti požadovanou část z disku do paměti;

poznamenej si, která část pole byla do paměti přečtena;

c. přepočti indexy, vyber prvek pole a pošli volajícimu programu jeho hodnotu.

Zápis nové hodnoty do pole je o něco složitější.

Poznamenávám, že uvedený trik je přímo zabudován do některých novějších operačních systémů.

### 3.4 Složitě algoritmy a co s nimi:

Z povahy věci vyplývá, že VTV mívají velmi složité algoritmy. Z dob, kdy ještě počítače pracovaly dosti pomalu se traduje, že program musí být napsán tak, aby výpočet byl co nejkratší. Chyba je v tom, že takto napsaný program bývá naprosto nečitelný. Klasikům to nevadilo: VTV se naprogramoval, provedl a pak se program uložil pro případný další výpočet v dalekém budoucnu.

Jakmile se VTV stává součástí úlohy nebo skupiny úloh ASŘ, musíme počítat s pozdějšími zásahy a úpravami a program i jeho dokumentace musí mít určitou formu (o tom dostatečně pojednávají sborníky z Havířova z let 1976-79). V celé programátorské praxi není dramatičtější situace než ta, do které je nyní doveden vedoucí pracovník: přimět vznešeného programátora VTV k psaní zřetelného, komentovaného, průhledného a čitelného programu; obzvláště silné vedoucí osobnosti navíc trvají na dobré dokumentaci. Je to však správné a nutné. Jestliže chceme produkovat programy, užitečné pro praxi, musíme na uvedených zásadách trvat, byť i to stálo přímo nadlidské úsilí. Programátorská šlechta si musí konečně zvyknout na fakt, že i nejchytřejší program, který není přehledný a nemá perfektní dokumentaci není k ničemu.

K tomu drobná rada: snažte se v programátorech VTV vzbudit zájem ("ty to neumíš napsat čitelněji?") a učinit jim z přehledného programování sport.

### 3.5 Prokleté GOTO:

V předcházejícím odstavci jsme konstatovali nutnost přehledného programování i těch nejsložitějších algoritmů.

V této souvislosti se jako přízrak vynořuje zákaz používání GOTO, tak často diskutovaný, vychvaňovaný i zatracovaný.

Při programování VTV bývá problém zjednodušen. Jestliže totiž chcí programovat bez GOTO, musím pracovat v jazyku, který umí alespoň

```
IF ... THEN ... ELSE ...
```

```
PERFORM (nebo DO ... WHILE ... a pod.)
```

a navíc by měl dovolovat blokovou strukturu (odstavce Cobolu nebo BEGIN .. END v Algolu a PL/1). Obvyklé implementace Fortranu a Basicu jsou v tomto směru chudé.

Nutno si ale uvědomit, že "program bez GOTO" je symbol pro určitou metodu, styl práce. Na příklad někdo napíše

```
IF (X-Y) 1Ø,2Ø
```

```
1Ø Z = Ø
```

```
GO TO 3Ø
```

```
2Ø Z = 1
```

```
3Ø .....
```

a někdo jiný totéž takto:

```
Z = Ø
```

```
IF (X .GE. Y) Z = 1
```

Ostatně překrásný příspěvek na toto téma je uveden ve sborníku z Havířova 1977 (Ing. Tvrdík: Poznámky ke struktuře fortranských programů). Lze tedy i ve Fortranu a Basicu dodržet styl práce, který měli na mysli autoři zákazu GOTO.

Závěrem: kromě dodržení základních pravidel čitelného programu a kromě běžné dokumentace měl by programátor VTV

- často používat komentářů
- v podprogramech (SUBROUTINE, FUNCTION) uvést vždy stručně funkci podprogramu a přesně popsat formální parametry
- v dokumentaci uvést seznam proměnných a výkladem
- popsat použitá pole, zvláště důkladně pak COMMON
- popsat všechny použité triky a finty.

## 4. Vědec a technik programují.

### 4.1 Možnosti a úskalí:

Řada vědců a techniků se naučila obstojně programovat v Basicu (případně i Algolu neb Fortranu). Někteří z nich se však nechali strhnout pocitem ovládnutí počítače a přehánějí v tom smyslu, že oni se zcela obejdou bez profesionálních programátorů. Tato tvrzení přišla vhod úředníkům, kteří se nyní pozastavují nad tím, proč vlastně existují programátoři VTV.

Pokusme se uvést věc na pravou míru; při programování VTV se vyskytují problémy, na př.

- u větších úloh bývá rozdíl mezi "algoritmem výpočtu" a "algoritmem počítačového zpracování"; na př. úlohy na grafu popíšeme snadno aparátem teorie množin; realizace na počítači vyžaduje znalost programátorské práce na seznamech, která je laikovi prakticky nedostupná;
  - úlohy s velkými bázemi dat vyžadují dobrou znalost práce se soubory; laik nemívá čas ani možnosti tuto dnes již rozsáhlou disciplinu studovat a prakticky zvládnout;
  - úlohy k opakovanému zpracování vyžadují programátorské zvládnutí provozních problémů počítače; laik bývá nakloněn k podceňování těchto problémů, což vede ke komplikacím v provozu;
  - laik se snaží hlavně o to, aby program pracoval; profesionál se zabývá i strukturalizací, modularitou, čitelností a pod.
- Na druhé straně je zapojení laiků do programovacích prací účelné a efektivní.

Situaci je nejlépe řešit vytvořením "programátorského servisu", skupiny špičkových programátorů VTV. Tato skupina konzultuje řešení úloh s laiky, sama řeší komplikovaná zadání a navíc provádí školení a metodický dohled.

Pokud však jde o programování VTV v rámci ASŘ, musí řešitelé být zásadně profesionální programátoři s dobrou praxí.

### 4.2 Kdo problému rozumí:

U specializovaných VTV se leckdy dostáváme do potíží při komunikaci programátora (řešitele) s vědci a techniky (jako zadavateli a uživateli). Ideálem je laik, který sám pro-

gramuje a programátor, který rozumí danému problému. Protože laiků se vztahem k výpočetní technice přibývá, redukuje se otázka na získání programátorů se znalostí příslušné vědeckotechnické specializace. Návod k řešení je jednoduchý, v praxi se obtížně realizuje: stabilizace programátorských kádrů a jejich vhodná výchova.

Při budování AŠH musí navíc programátor VTV komunikovat s ostatními řešiteli. Všichni vedoucí pracovníci by si měli uvědomit, že celkový výsledek řešitelských prací rozsáhlých systémů je vždy ve velké míře závislý na úrovni mezilidských vztahů.

## 5. Užitečné triky.

### 5.1 Modularita:

Důsledná výstavba programového systému na principu skládky modulů je dostatečně popsána v literatuře a byla podrobně rozebrána ve sbornících z Havířova (1976-79). I programy VTV mají být modulárně řešeny; zde na nás však číhají různé komplikace. Především je nutno důmyslně navrhnout struktury polí, zvláště pak COMMONS, aby jednotlivé moduly neměly problémy s přístupem k žádaným hodnotám. Svízeť bývá ve sklonu vědeckotechnických programátorů zaneřádit paměť spoustou matic, pokud možno obrovských. Zvláště minipočítače to nelibě nesou. Výchozí disk bývá v uložení polí na disk (odst.3.3) nebo v dynamické alokaci paměti; musíme však vytvořit subrutinu přístupu k takto řešeným polím, kterou jsou ostatní moduly povinny používat.

Pozorně též zkoumáme, zda některé řešené moduly nemají obecnější význam a použitelnost. Je-li tomu tak, vypipláme příslušný modul, opatříme dokumentací a nezištně nabídneme spolupracovníkům. Na druhé straně uvážíme, zda v programu VTV není možno využít již hotových rutin (z knihovny počítače nebo od pilných kolegů). Program VTV představuje náročnou a nákladnou práci a každá racionalizace je vítána.

## 5.2 Dialog člověk - stroj:

V řadě úloh VTV narážíme na problém nezvládnutelné složitosti výpočtu. V některé fázi řešení můžeme dospět do stavu, kdy nejame schopni předem určit, jaké skutečnosti další postup ovlivní. Program, který by postihoval všechny myslitelné možnosti byl by nesmírně komplikovaný a nákladný. Zde je výhodné přerušit práci počítače, zajistit výpis komplexních podkladů a předat problém k posouzení profesnímu odborníkovi. Ten zpracuje získané výsledky a rozhodne o způsobu pokračování. Další fázi řešení může opět převzít počítač.

Využitím principu dialogu zavádíme do procesu řešení užitečné prvky jako je "profesní cit odborníka". Podrobně o tom pojednává lit. /10/. Jestliže je dialog vhodně navržen, výrazně se zjednoduší programy a systém pracuje pružně, operativně a efektivně.

## 5.3 Zákoutí matematiky:

Dovolují si zde upozornit na neblahou vlastnost většiny vědců a techniků s "klasickým" matematickým vzděláním. Je to snaha řešit vše pomocí funkcí, derivací, integrálů, soustav rovnic a pod. Protože nenahlédli do tajů pro ně odlehlých partií matematiky (jako na př. množinové algebry), považují všechny disciplíny typu teorie grafů, her, algoritmů atd. za jakési šarlatánství, nehodné graduovaného vědce.

Přitom zmíněné partie matematiky umožňují často elegantně řešit úlohy, klasickými metodami nezvládnutelné. Vezměme typický případ - problém obchodního cestujícího (jak navštívit všechna zadaná místa při co nejkratší dráze s návratem do výchozího bodu). Řešení na př. pomocí hledání minima funkce je neuskutitelné; algoritmy, sestavené na základě teorie grafů pro řešení tohoto problému existují a fungují. Jiný příklad uvidíme v kapitole 6.

## 6. Zajímavý příklad.

### 6.1 VTV ve spojích:

Spoje se vám starají nejen o to, abyste mohli posílat dopisy svým přátelům (přítečkynám) nebo abyste se mohli vybavovat po telefonu (a obsazovat cenné spojovací okruhy). Spoje též zajišťují slušný poslech rozhlasových stanic a dopravu perfektního te-

levizního obrazu na obrazovky, před kterými trávíte tolik času. S touto činností je spojena řada problémů, z nichž většinu lze řešit jedině na počítačích. Ukážeme si jeden typický příklad.

### 6.2 Koordinace družicových spojů:

Na celosvětové úrovni probíhají přípravné práce pro řešení systému geostacionárních družic, vysílajících signál televize přímo k účastnickým přijímačům na zemi. Pro tento systém se na počítačích řeší řada úloh, na př.:

- a) Nalezení takového vysílacího svazku (v podobě eliptického kužele), jehož průnik se zemským povrchem vytvoří křivku, přibližující se co nejtěsněji k hranicím teritoria, které má družice obsluhovat. Počítač řeší transformaci hranic teritoria do jisté roviny a pak hledá ke vzniklému mnohoúhelníku elipsu s nejmenším obsahem.
- b) Každá družice vysílá z určité polohy na oběžné dráze na určitém knitočtu a s určitou polarizací. Tyto tři parametry ovlivňují velikost rušení, které družice způsobuje v "nepříslušných" teritoriích. Úloha přidělování knitočtů spočívá v nalezení hodnot všech tří parametrů pro každou družici (celkově se počítá se stovkami družic) tak, aby byly dodrženy omezující podmínky a minimalizovány určité ukazatele. Program používá dialogu člověk-stroj při návrhu poloh; pro určení knitočtu a polarizace vyvinul VÚS Praha algoritmus, opírající se o teorii grafů a využívající některých dalších poznatků teorie množin. Řešení všech problémů družicové sítě bylo skloubeno do jediného programového systému, který byl s úspěchem prakticky použit a jehož výsledky byly na mezinárodním poli vysoce oceněny.

### 6.3 Další aplikace:

Principy, ověřené při návrhu družicových spojů jsme dále rozpracovali a nyní je aplikujeme i v ostatních oblastech radiokomunikací. Současně řešíme otázky integrace systémů pro knitočtové plánování do ASŘ radiokomunikací.

Podrobnější údaje by případný zájemce našel v lit. /4/ a /5/ a v příslušné projektové dokumentaci.

## 7. Závěr.

Možná, že téma tohoto příspěvku bude leckomu připadat příliš odtažitě. Mám ale za to, že programátor má mít základní přehled i o problémech, se kterými se třeba nikdy nepotká.

V každém případě však věřím, že si na tento příspěvek vzpomenete za několik málo let, až budete sledovat přímé televizní vysílání z družice. Každého pravověrného programátora bude jistě hrát vědomí, že závratný pokrok, jehož je svědkem, se opírá o práci našich (přiznejme si to) milovaných počítačů.

## 8. Literatura.

- /1/ Bébr R.: Využití minipočítačů pro řízení a vědeckotechnické výpočty. Sborník SIAP 76, DT ČVTS Praha, 1976.
- /2/ Bébr R.: Využití počítačů při návrhu družicových spojů. Věd. konference ČVUT, sborník, ČVUT Praha, 1977.
- /3/ Bébr R.: Člověk rozalouvá se strojem. Sborník sem. Havířov, DT ČVTS Ostrava, 1977.
- /4/ Bébr R.: Nešystemnýj analíz i koordinirovanie radiosvjazi. II simposium po voprosam elektromagnitnoj sovmestivosti sputnikovyh i nasemyh sistem svjazi, Interkosmos - sborník VÚS Praha, 1978.
- /5/ Bébr R.: Praktické využití exaktních metod v některých spojových úlohách. PTT Revue č.3, 1978, NADAS.
- /6/ Knžera P.: Úloha kapesních počítačů a kalkulaček v automatizovaných systémech řízení. Dipl.práce VŠE Praha, 1978.
- /7/ Bébr R.: Nacionalizace projektování a programování při tvorbě ASŘ ve spojích. Zpravodaj ASŘ spojů, VÚS Praha 1979.
- /8/ Tvráček J.: Poznámky ke struktuře fortranových programů. Sborník sem. Havířov, DT ČVTS Ostrava, 1977.
- /9/ Berstiss A.T.: Data structures - Theory and Practice. Academic Press, New York 1975.
- /10/ Chytil M.K.: Forma a stupně uživatelské podpory při strojovém řešení kognitivních problémů. Sborník Programování 79, DT ČVTS Ostrava, 1979.