

Použitie nekonvenčných metód a vysokovýkonného počítania na riešenie rozvrhovacích problémov vo výrobe.

I. Budinská, J. Zelenka, T. Kasanický, Z. Balogh

Ústav informatiky SAV, Dúbravská cesta 9, 845 07 Bratislava, Slovakia,
e-mail: budinska.ui@savba.sk

Jednou z dôležitých úloh pri riadení výrobných podnikov je efektívne plánovanie a rozvrhovanie výroby. Už pri jednoduchej výrobnej linke pozostávajúcej z niekoľkých výrobných zariadení, úloha nájdenia optimálneho rozvrhu predstavuje NP-hard problém. Z praktického hľadiska nie je nutné nájdenie optimálneho riešenia, ale čo najlepšieho riešenia v čo najkratšom čase. K sľubne sa vyvíjajúcim metódam na riešenia rozvrhovacích úloh vo výrobe patrí napr. aj použitie multiagentových systémov, aplikácia PSO (Particle Swarm Optimization) algoritmu a umelých imunitných systémov. Ich použitie je v článku demonštrované na optimalizácii rozvrhovania úloh pre recyklačnú linku na výrobu plastových fólií. Dôležitou je požiadavka na potrebu častej zmeny rozvrhov pri príchode nových objednávok. Takéto zmeny v rozvrhu sa musia uskutočňovať rýchlo a súčasne je potrebné dodržať veľké množstvo obmedzení, ktoré vyplývajú aj z charakteru predchádzajúcich objednávok (niektoré nie je možné preplánovať). Skrátenie výpočtového času môže byť dosiahnuté aj použitím paralelných algoritmov, počítaním na klastroch, prípadne "cloudoch". Článok v závere rozoberá možnosti použitia vysokovýkonnej výpočtovej infraštruktúry a "cloud computingu" pre malé a stredné podniky.

Plánovanie a rozvrhovanie vo výrobe spolu s ďalšími manažérskymi úlohami, napríklad riadenie dodávateľsko-odberateľského reťazca patria aj v súčasnosti k zaujímavým úlohám výskumu a vývoja v tejto oblasti. Hranica medzi rozvrhovacími a plánovacími úlohami je veľmi neurčitá. Všeobecne sa dá povedať, že plánovanie sleduje viac kvalitatívne kritériá a robustnosť, zatiaľ čo rozvrhovanie sa sústreďuje na časové a nákladové kritériá, ako využiteľnosť zdrojov, zmenšenie prestojov, zníženie nákladov (napr. aplikovaním energeticky výhodnejších variánt rozvrhov. Do prípravy rozvrhov vstupuje veľké množstvo požiadaviek a obmedzení. Rozvrhovanie spravidla vedie k NP hard problémom, ktoré sa vyznačujú vysokou výpočtovou náročnosťou. Riešeniu rozvrhovacích úloh sa venuje množstvo vedeckých prác. Štandardné metódy nie sú vždy vhodným riešením práve kvôli dlhšej dobe výpočtu a kvôli zložitej implementácii algoritmov. Komplexita rozvrhovacích algoritmov často prekračuje možnosti bežne používaného hardvéru. Keďže v praxi nie je nutné nájsť optimálne riešenie, ale vhodné riešenie za primeraný čas a cenu, výskum v tejto oblasti sa orientuje na neštandardné metódy a algoritmy, napr. z oblasti umelej inteligencie. Veľmi sľubné sa ukazujú byť aj metódy založené na podobnosti s biologickými systémami. V tomto článku sa venujeme niektorým neštandardným metódam riešenia rozvrhovacích úloh s cieľom zvládnuť vysokú komplexitu výpočtov. Ďalšou možnosťou, ako sa vysporiadať s týmto problémom je využitie výkonného hardvéru a paralelných výpočtov. Vzhľadom na to, že obstarávacia cena vysokovýkonnej

výpočtovej techniky by sa premietla do ceny riešenia rozvrhovacích úloh, pre malé a stredné podniky by takéto riešenie neprineslo požadovaný efekt. Riešenie pre tieto podniky predstavuje nový trend v oblasti IT – tzv. cloud computing.

Úloha optimalizácie pre rozvrhovanie úloh vo výrobných systémoch

Všeobecne je úloha optimalizácie rozvrhovania vo výrobnom systéme (výrobná linka, výrobná dielňa) je definovaná ako vytvorenie optimálneho rozvrhu pre jednotlivé výrobné zariadenia, tak aby boli všetky objednávky vyrobené načas, minimalizovali sa prestoje na jednotlivých linkách, redukoval sa vznikajúci odpad a znížila sa spotreba elektrickej energie. Výsledkom by malo byť zvýšenie flexibility výroby a zníženie produkčných nákladov.

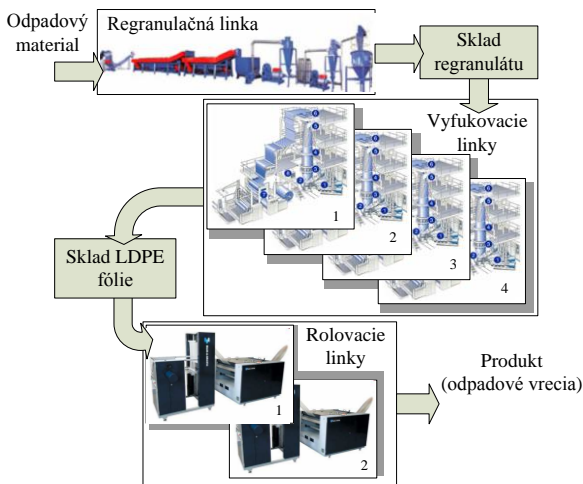
V súčasnosti je v mnohých malých a stredných podnikoch plánovanie výroby založené na intuitívnom rozhodovaní človeka. Takýto prístup je vhodný pre krátkodobé plánovanie, avšak pri väčšom množstve prijatých objednávok nie je možné pripraviť rozvrh na dlhšie obdobie dopredu. Ďalšie problémy vznikajú pri prijatí objednávok, ktoré vyžadujú preplánovanie výroby. Pri snahe dodržiavať veľké množstvo obmedzení vyplývajúcich z výrobného postupu, charakteristík výrobných zariadení a časových obmedzení, je ťažké uplatňovať ďalšie kritériá, ktoré by určitým spôsobom optimalizovali výrobu. Optimalizovanie vytváraných rozvrhov je možné za predpokladu existencie softvérovej aplikácia, ktorá bude pomáhať riadiacim pracovníkom, umožní im

simulovať viacero variánt rozvrhov a zvoliť ten, ktorý je pre nich z určitého hľadiska najvhodnejší.

Metódy a algoritmy prezentované v tomto článku boli testované na prípade reálnej výrobnéj linky na recykláciu plastov. Táto linka a výsledky simulácie v prostredí MATLAB Simulink boli popísané v článku [8]. Kvôli lepšej zrozumiteľnosti uvádzame aj tu stručný popis výrobnéj linky na recykláciu plastov.

Výrobný systém sa zaoberá recykláciou tzv. mäkkých plastov na báze polymerizácie a následnou výrobou odpadových vriec z LDPE fólie. Bloková schéma výrobnéj linky znázornená na obrázku 1 sa skladá z troch hlavných častí výrobného systému [8]:

- regranulačná linka: procesom polymerizácie odpadového materiálu dochádza k výrobe regranulátu rôznej farby (výroba regranulátu sa uskutočňuje tzv. mokrou cestou);
- vyfukovacia linka (extrúder): procesom polymerizácie regranulátu a ďalších vstupných aditív dochádza k výrobe LDPE fólie požadovaného tvaru, hrúbky, šírky a farby (k dispozícii sú štyri rôzne typy extrúderov);
- rolovacia linka: vyrobená LDPE fólia sa zvára, perforuje a roluje na požadovanú veľkosť (k dispozícii sú dve rolovacie linky).



Obrázok 1. Bloková schéma výrobného systému (AT&P 8/2010, id 109)

Pri hľadaní optimálneho rozvrhu pre jednotlivé časti výrobného systému je potrebné prehľadať celý priestor možných riešení. V prípade n úloh pre výrobu regranulátu $J_{gran,n}$ sa jedná o prehľadanie priestoru o veľkosti $n!$ kombinácií, avšak pri počte n úloh pre výrobu odpadových vriec $J_{vrec,n}$ sa tento priestor zväčší na $n!^2 \cdot 8^n$ kombinácií. Za predpokladu, že každá operácia sa môže vykonávať na každom stroji, definovanom pre danú operáciu. Ako možno vidieť prehľadávaný priestor je príliš veľký, napr. pre $n=2$ by sa jednalo o prehľadanie 256 kombinácií, čo má negatívny vplyv na výpočtovú náročnosť. Aplikovaním množiny precedencií na prehľadávaný priestor sa

redukuje prehľadávaný priestor. V našom prípade sa jedná o nasledovnú množinu precedencií:

$$\left\{ \begin{array}{l} o_{gran_i} < o_{vyf_i}; o_{vyf_i} < o_{rol_i}; o_{rol_i} < o_{bal_i}; \\ o_{gran_i} < o_{bal_i}; o_{vyf_i} < o_{bal_i}; o_{gran_i} < o_{rol_i} \end{array} \right\}$$

Pre riešenie problému rozvrhovania v reálnom výrobnom systéme je dôležitý čas výpočtu. Určitým zjednodušením optimalizácie by mohlo byť rozvrhovanie na úrovni množiny úloh. V takom prípade by sa prehľadávaný priestor redukoval na $n!$ kombinácií, čím sa redukuje požiadavky na výpočtový čas, ale za cenu nájdenia sub-optimálneho rozvrhu. Z tohto dôvodu sa budeme pohybovať iba v oblasti sub-optimálnych riešení.

Nekonvenčné metódy na rozvrhovanie

Na riešenie rozvrhovacích úloh sa používajú metódy a algoritmy, ktoré rodelujeme z hľadiska prehľadávania priestoru možných riešení na úplné (ktoré vždy nájdu optimálne riešenie, ak existuje) a na neúplné (pohybujú sa v oblasti suboptimálnych riešení). K neúplným metódam zaraďujeme aj neštandardné metódy, ako sú napríklad evolučné algoritmy, heuristické metódy, multiagentové systémy a podobne. V tomto článku sú charakterizované tri prístupy – multiagentové systémy, umelé imunitné systémy a PSO algoritmy.

Multiagentové systémy

Myšlienka na použitie multiagentových systémov pre riadenie podnikov má korene v 90-tych rokoch minulého storočia. Vzniklo viacero projektov, ktoré sa venovali návrhu architektúry multiagentových systémov a ich implementácii. Medzi najznámejšie takéto systémy patrí systém ProPlanT¹, ktorý vznikol pre potreby podniku TESLA-TV a je určený na plánovanie aktivít na všetkých úrovniach riadenia podniku. Pri vývoji tohto systému bol uplatnený trojúrovňový model riadenia podniku, kde na spodnej úrovni sa nachádzajú úlohy rozvrhovania úloh a operácií na jednotlivé stroje, teda ide o rozvrhovaciu úlohu takého typu, akej sa venujeme aj v tejto práci. Pri vytváraní modelu výrobnéj linky sme vychádzali z nasledujúcich predpokladov²:

Každú linku reprezentuje jeden agent a tieto agenty sú integrované pomocou plánovacieho agenta. Komunikáciu používateľa so systémom zabezpečuje administratívny agent.

Agenty môžu byť implementované v ľubovoľnom nástroji na impelmentáciu JAVA agentov.

K často používaným prostrediam patrí JADE, ktorý poskytuje API v jazyku Java, podporuje plný štandard

¹ www.proplant.cz

² Implementácia agentovej platformy na simuláciu výrobnéj linky bola urobená ako súčasť diplomového projektu: M. Jesenský -DP II, FIIT STU, 2010

FIPA a ako agentová platforma obsahuje nástroje (špeciálnych agentov, rozhrania, metódy) na integráciu agentov s inými technológiami ako sú SQL databázy alebo Java servlety.

Každý agent je charakterizovaný vlastnými parametrami, má vlastný cieľ a kriteriálnu funkciu na vyhodnocovanie optimálnosti. Komunikácia medzi agentami je zabezpečená posielaním FIPA správ. Na ovládanie MAS používateľom slúži webové rozhranie. Komunikácia prebieha medzi administratívnym agentom a plánovacím agentom. Komunikáciu riadi a iniciuje používateľ systému cez webové rozhranie. Administratívny agent vybuduje príslušné správy pre agentov, odošle ich a prijme odpovede, ktoré vráti webovému rozhraniu.

Plánovací agent je iniciátorom vnútornej komunikácie agentov pri spracovávaní nových objednávok a tvorbe rozvrhov. Táto komunikácia nemá pre používateľa žiadny výstup. Všetky interakcie používateľa so systémom a výstupy systému riadi administratívny agent spolu s webovým rozhraním.

Zadávanie nových objednávok je realizované komunikáciou administratívneho agenta s plánovacím, ktorý si ich ukladá do pamäte a neskôr ich hromadne spracuje na úlohy, ktoré priradí do rozvrhov. Nevýhodou tohto spôsobu je, že plánovací agent stráca objednávky po ich spracovaní. Pre potrebu rozšírenia systému o správu objednávok a výpočty štatistik je potrebné načítanie objednávok z exportu údajov zo systému. V tejto práci ale štatistiky výroby nie je nepotrebné vypočítavať, preto postačuje realizácia systému bez databázy.

MAS simuluje výrobný proces, preto má každý agent svoj stav odvodený od aktuálneho času a používateľ môže zadať vykonanie jedného kroku výroby. Jeden krok predstavuje zmenu času a stavu všetkých agentov podľa najbližšej udalosti niektorého agenta od aktuálneho času.

Evolučné algoritmy a umelé imunitné systémy

Evolučné algoritmy sú založené na mechanizme, ktorý napodobuje evolciu biologických systémov: reprodukcia, mutácia, rekombinácia a selekcia. Pri riešení rozvrhovacích úloh, možné rozvrhy predstavujú individual v populácii. Nevýhodou použitia evolučných algoritmov je problém so správnym nastavením parametrov a problém s transformáciou problému na chromozómovú reprezentáciu. V literatúre [1][2] sa popisujú dva základné prístupy k reprezentácii problému rozvrhovania typu job-shop:

- Priama chromozómová reprezentácia. Problém rozvrhovania je reprezentovaný tak, že jednotlivé rozvrhy predstavujú chromozómy Táto metóda vyžaduje vyvinutie špecifických genetických operátorov.
- Nepriama chromozómová reprezentácia. Chromozómy neprezentujú priamo rozvrh.

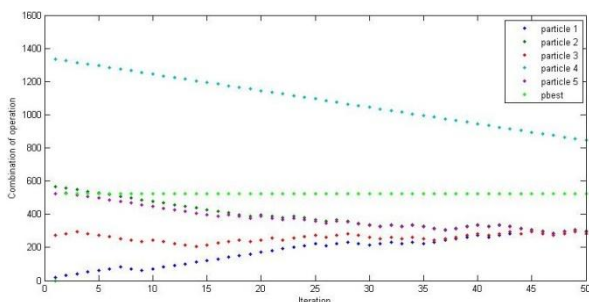
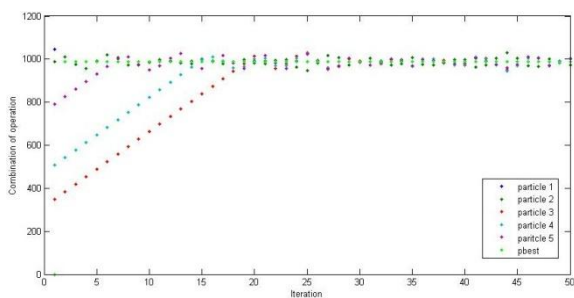
V evolučnom prístupe je potrebné nastaviť veľké množstvo konštánt a je veľmi dôležité nastaviť ich korektné, tiež je často nevyhnutné vyvinúť nové genetické operátory pre špecifickú reprezentáciu problému. Výsledok stochastického hľadania závisí od fitness funkcie. Fitness funkcia je zodpovedná za optimálnu distribúciu riešení vo vyhľadávacom priestore. Fitness funkcia môže reprezentovať model všetkých zdrojov (strojov), ktoré sa využívajú v rozvrhu. Niekedy je ťažké nájsť vytvoriť vhodný model strojov.

Skupina algoritmov motivovaná imunitným systémom cicavcov patrí medzi nové prístupy testované na doméne rozvrhovania a plánovania. Oblasť umelých imunitných systémov (artificial immune systems AIS) je relatívne mladá, v súčasnosti sa tieto algoritmy úspešne začínajú používať v rôznych aplikáciách, napríklad aj pre riešenie optimalizačných problémov. Pre nájdenie optimálneho riešenia je vhodný najmä princíp klonálnej selekcie. Funkcia klonálna selekcia imunitných systémov sa dá interpretovať ako špeciálny mikrokozmos Darwinových zákonov evolúcie s tromi hlavnými princípmi repertoárovej diversity, genetickej variácie a prírodnej selekcie [3]. Algoritmus CLONALG, ktorý je motivovaný aktiváciou B buniek imunitného systému pri napadnutí patogénnou látkou, sa často používa aj pri riešení optimalizačných problémov. Práce [4][5][6] ukazujú, že algoritmus CLONALG je užitočnou alternatívou k štandardným riešeniam pre job shop rozvrhovanie. Hlavný rozdiel medzi štandardným riešením evolučnou schémou a algoritmom CLONALG je, že CLONALG nemá žiadne rodové operátory. Uplatňuje sa tam len propocionálna mutácia, takže keď individuum je blízko riešeniu, pravdepodobnosť mutácie je nízka. Na druhej strane, keď afinita individua k riešeniu je nízka, pravdepodobnosť mutácie stúpa. Pretože CLONALG nemá rodové operátory, konvergencia je pomalšia ako v prípade evolučných prístupov, ale je väčšia šanca nájsť globálne optimum.

Podstatu algoritmu CLONALG tvorí pseudo evolučný cyklus, ktorý sa však neopiera o rodové operácie, ale posúva sa v priestore riešení iba na základe progresívnej mutácie. Samotné riešenie je založené na fakte, že bunka je reprezentovaná maticou, ktorá popisuje súslednosť jednotlivých objednávok, na jednotlivých strojoch v čase. Afinita jednotlivých buniek je rátaná na základe vzorca, ktorý v sebe zahŕňa všetky požiadavky kladené na optimálne riešenie. To znamená, že ak je afinita bunky vysoká tak matica objednávok na jednotlivých strojoch reprezentuje riešenie blízke optimálnemu riešeniu [7]. Počas testovania sa ukázalo, že zvolený algoritmus zvláda zadanú úlohu. Proti konvenčným metódam založeným na plnom prehľadávaní priestoru tento prístup ponúka rýchlejšie nájdenie suboptimálneho riešenia pri väčšom počte objednávok, ale nezaručuje nájdenie optimálneho riešenia.

PSO

Ďalšou možnosťou zo skupiny biologicky motivovaných algoritmov je algoritmus Particle swarm optimization – PSO. PSO algoritmus patrí do skupiny heuristických metód, pretože ani tento algoritmus nezaručuje nájdenie optimálneho riešenia. Jeho výhody sa prejavujú pri hľadaní optimálneho riešenia v priestore s veľkým počtom možných kombinácií. Algoritmus zamestnáva populáciu prehľadávacích bodov, označovaných ako častice „particle“, ktoré sa stochasticky posúvajú v prehľadávanom priestore. Z jednej populácie „swarm“ sa vyberie najlepšia pozícia z ktorej sa odvodzuje ďalší smer jednotlivých častíc v prehľadávanom priestore. Po určitom počte iteračných krokov sa jednotlivé častice začnú zhromažďovať v oblasti optimálneho riešenia. Následne sa tieto častice náhodne pohybujú v okolí tohto nájdeného riešenia a zisťujú, či sa v jeho okolí nenachádza lepšie riešenie. Ukončenie prehľadávania môže byť zabezpečené splnením cieľovej optimalizačnej funkcie alebo počtu iteračných krokov. Na obr. 2 a), a b) v [9] možno sledovať výsledok pohybu jednotlivých častíc v prehľadávanom priestore a taktiež ich schopnosť zhromažďovať sa v oblasti optimálneho riešenia.



Obr. 2 Pohyb častíc prehľadávacím priestorom a ich zhromažďovanie v oblasti optimálneho riešenia

Rýchlosť nájdenia optimálneho riešenia je závislá od veľkosti prehľadávacieho priestoru, od nastavenia parametrov ovplyvňujúcich výpočet smeru a vzdialenosti jednotlivých častíc a cieľovej optimalizačnej funkcie. Keďže nie vždy je možné vhodne nastaviť resp. určiť cieľovú optimalizačnú funkciu, ukončenie algoritmu bude zabezpečené pevným nastavením počtu iteračných krokov. Pre túto hodnotu sa budú prispôbovať aj parametre ovplyvňujúce výpočet smeru a vzdialenosti jednotlivých častíc. Nájdenie

optimálneho riešenia závisí od dĺžky výpočtu. Skrátenie dĺžky výpočtu a tým aj zvýšenie pravdepodobnosti nájdenia optimálneho riešenia je možné napr. použitím paralelných výpočtov.

Paralelné výpočty – *Parallel computing toolbox* MATLAB

Keďže čas hľadania optimálneho, resp. Suboptimálneho riešenia, ako aj samotná cena tohto riešenia je dôležitým faktorom, ktorý rozhoduje o ochote implementovať tieto metódy do praxe vo výrobných podnikoch, najmä pre malé a stredné podniky, skúmame možnosti použitia paralelných výpočtov na zníženie výpočtového času. Prezentované metódy – PSO a AIS sú svojim charakterom vhodné na aplikáciu v paralelnom výpočtovom prostredí.

Programové prostredie Matlab poskytuje podporu pre paralelné výpočty ako na lokálnom počítači, tak aj na počítačových klastroch. Toolbox Parallel Computing umožňuje súbežné vykonávanie až ôsmich MATLAB implementácií na jednom lokálnom počítači. Ďalšou možnosťou je použitie distribuovaného výpočtového servera MATLAB. Tento server umožňuje prácu na vzdialených klastroch počítačov. Pre zbiehanie úloh na tomto serveri musí mať každý pracovník samostatnú desktopovú licenciu pre prostredie MATLAB. Softvér Parallel Computing Toolbox umožňuje použitie paralelných výpočtov nasledujúcimi spôsobmi:

- `parallel for-loops (parfor)`: je užitočný pre situácie, kde sa vyžaduje veľké množstvo iterácií v slučke pre jednoduché výpočty. Použitie príkazu `parfor` je podobné štandardnému príkazu `for` s tým rozdielom, že príkaz `parfor` rovnomerne distribuuje výkonomi medzi jednotlivé výpočtové uzly,

- `single program multiple data (spmd)`: umožňuje bezproblémové prelínanie sériového a paralelného programovania. `spmd` zabezpečuje vykonávanie identických kódov na viacerých výpočtových uzloch. Môžete napr. spúšťať jeden program k klientskej aplikácii MATLAB a tie časti programu, ktoré sú označené pomocou `spmd` do bloku, sa spúšťajú na vysokovýkonných výpočtových zariadeniach. Po ich ukončení program pokračuje vo vykonávaní programu sériovým spôsobom,

- paralelné operácie na klastroch – pozostávajú z jednej úlohy, ktorá beží súčasne na viacerých výpočtových uzloch simultánne. Jednotlivé úlohy obvyčajne používajú rôzne dáta.

Výhodou prvých dvoch spôsobov je, že môžu bežať na lokálnom počítači. Vyžadujú však zložitejšiu implementáciu najmä v súvislosti s odovzdávaním hodnôt premenných. Tretí spôsob nie je náročný na splnenie podmienok pre premenné, ale komunikácia medzi lokálnym zariadením a klastrom môže trvať

dlhšiu dobu, ako komunikácia v rámci jedného lokálneho zariadenia.

Ďalšou nevýhodou pre praktické použitie týchto prístupov je vysoká obstarávacia cena hardvérovej infraštruktúry potrebnej pre paralelné počítanie. Túto nevýhodu môže odstrániť používanie cloud computingu.

Cloud computing pre malé a stredné podniky

V poslednom čase sa veľa hovorí o „cloudoch“ (oblaky) a „cloud computingu“ (oblakové počítanie). Keďže sme sa doposiaľ nestretli s vhodným slovenským ekvivalentom tohto pojmu, budeme používať anglický názov. „Cloud computing“ zjednodušene povedané je technológia, ktorá umožňuje zdieľanie hardvérových a softvérových zdrojov prostredníctvom internetu. Služby a softvérové zdroje sú uložené na vzdialených serveroch a používatelia k nim môžu pristupovať na vyžiadať cez webové rozhranie, alebo priamo cez rozhranie aplikácie z ľubovoľného miesta na sieti. Výhodou používania cloud computingu najmä pre malé a stredné podniky je dostupnosť výpočtovej kapacity a softvérových riešení bez nákladných investícií do vlastnej infraštruktúry, licencií softvéru a zaškolenia pracovníkov. Pri používaní cloud computingu sa neplatí za vlastný hardvér alebo softvér, ale za jeho použitie. Takto je možné škálovať výpočtovú kapacitu a meniť požadovanú funkcionálnosť podľa aktuálnych požiadaviek a potrieb klientov.

Pre cloud computing je charakteristické:

- Služby a dáta sú umiestnené na vzdialených serveroch patriacich poskytovateľovi služby
- Služby a dáta sú dostupné z ľubovoľného miesta na sieti
- Poskytované služby a výpočtová kapacita sú škálovateľné – užívateľ platí len za skutočne potrebné služby a výkon

Okrem poskytovania softvéru a výpočtového výkonu, existujú aj poskytovatelia dátových úložísk. Veľké internetové firmy, ako Amazon, Google a pod. zistili, že využívajú len malú časť kapacity vlastných dátových skladov. Technológia cloud computingu umožňuje vytvoriť tzv. „cloud storage“ a prenajímať voľnú časť skladovej kapacity iným užívateľom prostredníctvom internetu. Medzi najznámejšie takéto služby v súčasnosti patria Amazon Elastic Compute Cloud (EC2) a Simple Storage Solution (S3).

V súvislosti s rozvojom sémantického webu sa diskutuje aj o možnosti využitia cloud computingu na ukladanie štruktúrovaných dát. V budúcnosti by využívanie cloud computingu mohlo výrazne prispieť k rozvoju malých a stredných firiem.

Záver

Komplexita pri riešení rozvrhovacích úloh vo výrobných systémoch predstavuje kritickú bariéru pre implementáciu rozvrhovacích metód a optimalizáciu

výkonu výrobných systémov. Použitie nekonvenčných metód, paralelných algoritmov a cloud computingu môžu výrazným spôsobom prispieť k zvýšeniu efektivity výroby a zisku výrobných podnikov.

Táto práca bola podporená agentúrou ASFEU v rámci projektu Podpora budovania centra excelencie pre Smart systémy, technológie a služby, ITMS 26240120005

Literatúra

[1] Baghi S., Uckun S., Miyab Y. and Kawamura K. (1991): Exploring problem-specific recombination operators for job shop scheduling. — Proc. 4-th Int. Conf. Genetic Algorithms, University of California, San Diego, pp. 10–17, July 13–16.

[2] Bruns R. (1993): Direct chromosome representation and advanced genetic operators for production scheduling. — Proc. 5-th Int. Conf. Genetic Algorithms, University of Illinois at Urbana-Champaign, pp. 352–359.

[3] G. Cziko, "The Immune Systems: Selection by the Enemy", in Without Miracles, G. Cziko, A. Bradford Book, The MIT Press 1995, pp. 39-48

[4] Hong Lu; Jing Yang; , "An Improved Clonal Selection Algorithm for Job Shop Scheduling," Intelligent Ubiquitous Computing and Education, 2009 International Symposium on , vol., no., pp.34-37, 15-16 May 2009
doi: 10.1109/IUCE.2009.26

[5] Hong, L. 2009. Stretching Technique-Based Clonal Selection Algorithm for Flexible Job-shop Scheduling. In Proceedings of the 2009 international Conference on Computational intelligence and Natural Computing - Volume 02 (June 06 - 07, 2009). CINC. IEEE Computer Society, Washington, DC, 111-114. DOI=<http://dx.doi.org/10.1109/CINC.2009.237>

[6] Jin-hui Yang, Liang Sun, Heow Pueh Lee, Yun Qian, Yan-chun Liang, Clonal Selection Based Memetic Algorithm for Job Shop Scheduling Problems, Journal of Bionic Engineering, Volume 5, Issue 2, June 2008, Pages 111-119, ISSN 1672-6529, DOI: 10.1016/S1672-6529(08)60014-1.

[7] KASANICKY, Tomas. Agentový systém na báze klonalnej selekcie. In Cybernetics and

Informatics 2010 : international conference SSKI. Editor Stefan Kozak, Alena Kozakova, Danica Rosinova. - Bratislava : Vydavateľstvo STU, 2010, 6 p. ISBN 978-80-227-3241-3

[8] Zelenka. J., Budinská. I., Balogh. Z., Matejka. T.: Problematika rozvrhovania vo výrobnom procese s rôznymi typmi výrobných liniek, AT&P journal 8/2010, dostupné na www.atpjournals.sk (august 2010, id článku: 109), ISSN 1336-233X

[9] Zelenka, J. (2010a), Application of Particle Swarm Optimization in Job-Shop Scheduling problem in the Recycling Process, CINTI 2010, Budapešť 2010, ISBN 978-4244-9278-7, IEEE CN CFP1024M-PRT, pp. 137-140