

UNIVERSITATEA PETROL-GAZE PLOIEȘTI
DEPARTAMENTUL AUTOMATICĂ, CALCULATOARE ȘI ELECTRONICĂ
STUDII UNIVERSITARE DE DOCTORAT

TEZĂ DE DOCTORAT

CONTRIBUȚII PRIVIND DEZVOLTAREA
UNUI SISTEM INTELIGENT DE
PLANIFICARE AUTOMATĂ A PRODUCȚIEI
ÎN LINII FLEXIBILE DE FABRICAȚIE

Conducător științific:

Prof. univ. dr. ing. Mihaela Oprea

Doctorand:

Chircu (Toader) Florentina Alina

2017

CUPRINS

Lista abrevierilor	5
Index figuri.....	7
Index tabele	10
Introducere.....	12
1. Planificarea automată a producției.....	17
1.1. <i>Formularea problemei de planificare a producției.....</i>	<i>18</i>
1.2. <i>Particularități ale planificării producției în liniile flexibile de fabricație.....</i>	<i>20</i>
1.3. <i>Viziunea sistemică asupra planificării automate a producției.....</i>	<i>24</i>
1.4. <i>Probleme de planificare.....</i>	<i>26</i>
1.4.1. <i>Planificare pe o singură mașină (OMP).....</i>	<i>26</i>
1.4.2. <i>Planificare pe mașini paralele (PMP).....</i>	<i>26</i>
1.4.3. <i>Planificare în flux continuu (FSP).....</i>	<i>27</i>
1.4.4. <i>Planificare a producției multisortimentale (JSS).....</i>	<i>28</i>
1.4.5. <i>Planificare a producției multisortimentale fără relație de precedență (OS).....</i>	<i>29</i>
1.4.6. <i>Exemplu de problemă de planificare în linii flexibile de fabricație.....</i>	<i>30</i>
1.5. <i>Modelul matematic al problemei JSS.....</i>	<i>35</i>
1.6. <i>Metode și algoritmi pentru planificarea producției.....</i>	<i>37</i>
1.6.1. <i>Metode clasice.....</i>	<i>37</i>
1.6.2. <i>Metode bazate pe inteligență artificială.....</i>	<i>38</i>
1.6.3. <i>Alte metode de planificare.....</i>	<i>45</i>
1.7. <i>Sisteme de planificare automată a producției în linii flexibile de fabricație.....</i>	<i>48</i>
1.7.1. <i>Sisteme dezvoltate pe plan national.....</i>	<i>48</i>
1.7.2. <i>Sisteme dezvoltate la nivel international.....</i>	<i>54</i>
1.8. <i>Concluzii parțiale.....</i>	<i>62</i>
2. Tehnici de inteligență artificială computațională pentru planificarea automată a producției în linii flexibile de fabricație.....	63
2.1. <i>Algoritmi Genetici.....</i>	<i>64</i>
2.2. <i>Particle Swarm Optimization.....</i>	<i>73</i>
2.3. <i>Ant Colony Optimization.....</i>	<i>82</i>
2.4. <i>Călirea simulată (Simulated Annealing).....</i>	<i>90</i>
2.5. <i>Stadiul actual al cercetărilor privind aplicarea algoritmilor IA pentru problema JSS98</i>	
2.6. <i>Metode hibride bazate pe IA.....</i>	<i>102</i>

2.7. Concluzii parțiale.....	105
3. Cercetări experimentale privind identificarea unei metode hibride eficiente de rezolvare a problemei de planificare JSS	106
3.1. Analiza comparativă a tehnicilor de Inteligență Artificială Compuțională pentru rezolvarea problemei JSS.....	106
3.1.1. Studiu de caz 1 – Set de benchmark-uri.....	107
3.1.2. Studiu de caz 2 – Set de date de test fictive.....	113
3.1.3. Concluziile analizei comparative.....	119
3.2. Metoda hibridă HibriPSO-SA – prezentare algoritm.....	121
3.3. Validarea algoritmului HibriPSO-SA.....	128
3.4. Concluzii parțiale.....	133
4. Dezvoltarea sistemului inteligent de planificare automată HibriPlanJSS bazat pe metoda hibridă propusă.....	135
2.1. Metodologia utilizată pentru dezvoltarea sistemului de planificare.....	135
2.2. Structura sistemului inteligent de planificare.....	136
2.3. Modelarea cu rețele Petri.....	137
4.1.1. Etapa de sinteză descendentă.....	138
4.1.2. Etapa de sinteză ascendentă.....	142
4.1.3. Modelarea rețelei Petri în PIPE v.4.3.0.....	144
2.4. Implementarea sistemului inteligent de planificare automată a producției.....	152
4.1.4. Studiu de caz – Unilever România.....	152
4.1.5. Descrierea sistemului HibriPlanJSS.....	156
2.5. Testarea algoritmului HibriPSO-SA.....	157
2.6. Analiza comparativă a metodei propuse pentru Unilever.....	159
2.7. Concluzii parțiale.....	164
Concluzii finale și direcții viitoare de cercetare.....	166
Lista de lucrări publicate	172
Bibliografie	174
Webografie	185
Anexa 1. Glosar de termeni	187
Anexa 2. Sinteza studiului sistemelor de planificare	189
Anexa 3. Modelarea cu rețele Petri.....	194

Anexa 4. Etapa de validare a metodei.....	204
Anexa 5. Procesul de fabricație a margarinei	211
Anexa 6. Codul sursă al aplicației	216
Anexa 7. Testarea metodei hibride	222

INTRODUCERE

Dezvoltarea unor sisteme de fabricație eficiente, adaptate la provocările economiei globale, din ce în ce mai dinamică, necesită noi tehnologii și instrumente software pentru optimizarea planificării automate a producției.

Cercetările realizate în ultimii ani s-au concentrat pe identificarea unor metode bazate pe tehnici de Inteligență Artificială Compuțională (IAC) din categoria Swarm Intelligence (Inteligența colectivă) care pot găsi soluții optime ale problemelor de planificare, probleme recunoscute ca fiind NP-dure cu o complexitate computațională foarte mare.

Planificarea automată a producției presupune descrierea etapelor de producție, ținând cont de materia primă și mașinile disponibile, de tipul de produse și de timpii de prelucrare asociați fiecărei mașini, fără a necesita intervenția operatorului uman.

Problema JSS (Job Shop Scheduling) este una dintre cele mai dificile din acest domeniu întrucât soluția acesteia trebuie să furnizeze un plan care oferă o maximizare a capacității de producție, precum și minimizarea timpului total de producție, împreună cu reducerea costurilor asociate și îmbunătățirea gradului de utilizare a resurselor disponibile.

În această lucrare vom aborda rezolvarea problemei JSS în linii flexibile de fabricație și vom propune o metodă hibridă bazată pe tehnici de Inteligență Artificială (IA), combinând o tehnică Swarm Intelligence cu o altă tehnică inteligentă, metodă validată pe un set de benchmark-uri și pe o linie flexibilă de fabricație de la Unilever Ploiești România.

Pentru acest domeniu atenția cercetătorilor s-a concentrat pe căutarea de soluții potrivite pentru crearea unor medii de producție capabile să facă față noilor cerințe, accentul punându-se pe identificarea unor produse inteligente care să ofere soluții optime pentru această problemă.

Importanța dovedită a realizării unei planificării optimale a dus la abordarea în diferite maniere a problemei JSS. Pe lângă algoritmi clasici asociați domeniului planificării, au fost propuse instrumente software bazate pe alte tipuri de algoritmi care

oferă o optimizare a anumitor termeni ce caracterizează planificarea producției (timpul total al execuției, gradul de utilizare a mașinilor, optimizarea utilizării resurselor, etc.).

Tendențele actuale ale cercetării în domeniul planificării optimale în linii flexibile de fabricație sunt orientate către adoptarea unor tehnici hibride, bazate pe algoritmi specifici inteligenței artificiale, care încearcă să îmbine avantajele strategiilor de căutare locală (cum ar fi Optimizarea prin modelul coloniilor de furnici – Ant Colony Optimization, Optimizarea prin modelul ansamblurilor de particule – Particle Swarm Optimization, Modelul roiurilor de albine – Artificial Bee Colony, Algoritmi genetici) cu cele ale căutării globale (cum ar fi Călirea simulată – Simulated annealing, Algoritmul de căutare bazat pe liste tabu – Tabu Search) pentru determinarea unor soluții optime și eficiente în cazul problemei propuse.

Obiectivul principal al cercetării doctorale a constat în identificarea unei metode de planificare eficientă a producției multisortimentale în linii flexibile de fabricație (prin minimizarea timpului total necesar execuției) și dezvoltarea unui sistem inteligent de planificare automată a producției în linii flexibile de fabricație, bazat pe metoda propusă.

Obiectivele specifice ale tezei de doctorat au vizat următoarele aspecte:

- Prezentarea stadiului actual al cercetărilor în domeniul planificării automate a producției;
- Adaptarea modelului matematic generalizat pentru problema Job Shop Scheduling (JSS);
- Analiza comparativă a eficienței algoritmilor de planificare (atât clasici, cât și specifici IA) pentru problema propusă;
- Identificarea și dezvoltarea unei metode hibride pe baza rezultatelor studiului comparativ;
- Validarea metodei hibride propuse pe baza unui set de benchmark-uri clasice;
- Testarea metodei hibride propuse pentru un sistem de fabricație real din cadrul Unilever România.

Teza de doctorat este structurată astfel:

- *Introducere*, în care este prezentat succint domeniul planificării automate, precum și scopul, obiectivele și structura acestei teze;
- Capitolul 1, *Planificarea automată a producției*, include formularea detaliată a problemei planificării, inclusiv particularitățile condiționate de domeniul liniilor flexibile de fabricație, prezentarea tipurilor clasice de probleme, dar și a metodelor și algoritmilor utilizați în acest domeniu. De asemenea, este propusă o viziune sistemică asupra planificării automate în liniile flexibile de fabricație, descrierea detaliată a modelului matematic al problemei Job Shop Scheduling (JSS), iar în ultima secțiune a acestui capitol este prezentată o analiză comparativă a sistemelor de planificare dezvoltate la nivel național și internațional.
- Capitolul 2, *Tehnici de inteligență artificială computațională pentru planificarea automată a producției în linii flexibile de fabricație*, include o prezentare succintă a tehnicilor de Inteligență Artificială Computațională studiate: Algoritmi Genetici, Particle Swarm Optimization, Ant Colony Optimization și Simulated Annealing. Cea de-a doua parte a acestui capitol conține o prezentare a stadiului actual al cercetărilor privind aplicarea tehnicilor specifice Inteligenței Artificiale pentru rezolvarea problemei JSS, precum și o sinteză a metodelor hibride propuse în literatura de specialitate pentru această problemă.
- Capitolul 3, *Cercetări experimentale privind identificarea unei metode hibride eficiente pentru planificarea în contextul JSS*, cuprinde o analiză comparativă a performanței tehnicilor de IAC pentru rezolvarea problemei JSS, prezentarea metodei hibride propuse, precum și validarea acesteia utilizând un set de benchmark-uri clasice, recunoscute în literatura de specialitate.
- Capitolul 4, *Dezvoltarea sistemului de planificare HibriPlanJSS bazat pe metoda hibridă propusă*, include detalierea metodologiei utilizate pentru dezvoltarea sistemului de planificare HibriPlanJSS, împreună cu structura generală și descrierea acestuia. Tot în această secțiune este inclus un studiu de caz realizat pe baza unui sistem de fabricație real, disponibil la Unilever

România. Acest sistem de fabricație este modelat cu rețele Petri pentru a oferi o imagine de ansamblu asupra modului de utilizare a resurselor comune și de funcționare a acestuia. În ultima parte a acestui capitol este inclusă etapa de testare a sistemului de planificare, precum și o analiză comparativă detaliată a rezultatelor obținute.

- *Concluzii finale și direcții viitoare de cercetare* evidențiază contribuțiile tezei de doctorat și trasează direcțiile viitoare de cercetare.

Mulțumiri

Domnișoarei prof. dr. ing. Mihaela Oprea, în calitate de conducător științific, aș vrea să îi adresez cele mai calde mulțumiri pentru încredere, pentru răbdarea cu care m-a îndrumat pe parcursul acestei colaborări și pentru tot sprijinul acordat.

Mulțumesc doamnei prof. dr. ing. Otilia Cangea pentru răbdarea și sprijinul acordate în elaborarea acestei teze de doctorat.

Mulțumesc domnului ing. Gabriel Oprea din cadrul Unilever România pentru colaborarea în realizarea studiului de caz din cadrul acestei teze de doctorat.

Respectul și mulțumirile mele se îndreaptă către membrii colectivului Departamentului Automatică, Calculatoare și Electronică pentru sfaturile și recomandările oferite. Mulțumesc de asemenea și colegilor din Departamentul Informației, Tehnologia Informației, Matematică și Fizică pentru susținerea și ajutorul acordat pe parcursul elaborării tezei.

Mulțumesc din suflet părinților pentru susținerea permanentă și prietenilor pentru încurajările primite în tot acest timp.

Mai mult decât simple mulțumiri adresez soțului meu, Alexandru, și fiului meu, Matei, care mi-au dat forța și optimismul necesar pentru a finaliza această teză.

1 PLANIFICAREA AUTOMATĂ A PRODUCȚIEI

Planificarea automată a producției constă în identificarea și detalierea tuturor etapelor necesare procesului de producției, luând în considerare stocul de materie primă disponibil, structura sistemului de producție, precum și tipurile de produse, fără a necesita intervenția continuă a operatorului uman [27].

Aceasta implică două etape diferite, ce se influențează reciproc [121]:

- *planificare (planning)* – procesul de identificare a tuturor activităților necesare pentru a finaliza un proiect;
- *ordonanțare (scheduling)* – determinarea ordinii secvențiale pentru fiecare activitate în parte, atribuirea timpului total necesar planului, precum și stabilirea datei de începere și de finalizare a acestuia.

VIZIUNEA SISTEMICĂ ASUPRA PLANIFICĂRII AUTOMATE A PRODUCȚIEI

Sistemul de planificare a producției asigură suportul pentru activitățile de programare și urmărire a acesteia. Pe baza capacității de producție și în funcție de preconizarea comenzilor se concep planuri de producție pe termen scurt, mediu sau lung, care includ sarcinile și termenele concrete pentru realizarea produsului finit [38].

În **Figura 1.3** este prezentată viziunea sistemică pe care autoarea o propune pentru problema planificării automate a producției.

Intrările sistemului de planificare sunt reprezentate de parametrii algoritmului de planificare implementat (parametrii specifici algoritmului de planificare, structura sistemului, caracteristicile mașinilor, caracteristicile produselor, detaliile privind fluxul tehnologic al produselor etc).

Ieșirea sistemului de planificare este reprezentată de o serie de specificații de planificare a operațiilor pe mașini (ordinea accesării mașinilor de către fiecare produs în parte, timpii de accesare și eliberare ai mașinilor etc) care sunt concretizate într-un plan

detaliat de execuție, precum și de timpul total necesar acestuia. Aceasta reprezintă intrările pentru procesul de fabricație, ieșirile acestuia fiind reprezentate de totalitatea produselor finite obținute prin respectarea specificațiilor planului de execuție (P_1, P_2, \dots, P_n).

Perturbațiile specifice procesului de fabricație sunt similare cu cele ale sistemului de planificare și sunt reprezentate de stadiul aprovizionării cu materie primă, de eventualele defecțiuni apărute de-a lungul sistemului de fabricație și de gradul de ocupare al magaziiilor de produse finite.

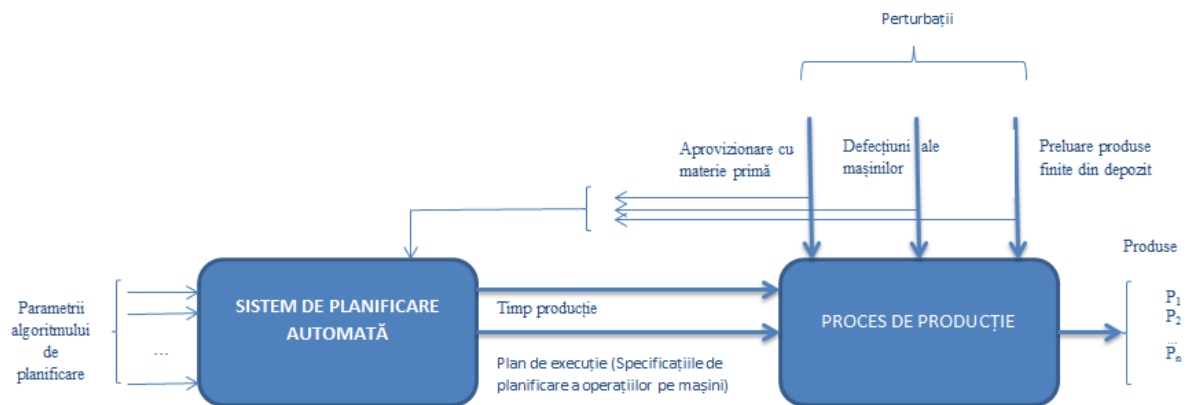


Figura 1.3 Structura generală a unui sistem de planificare automată

MODELUL MATEMATIC AL PROBLEMEI JSS

Întrucât în această teză vom aborda rezolvarea problemei JSS, prezentăm modelul matematic al acesteia. Problema JSS este una dintre cele mai greu de rezolvat probleme din mediul industrial, fiind inclusă în categoria celor NP-complexe, conform lui Srinivas și Allada în 2004 [132]. În aceste condiții, identificarea unui model matematic care să descrie cât mai bine toate detaliile problemei este esențială.

Modelul matematic al problemei JSS poate fi descris după cum urmează:

- $LM = \{M_1, M_2, \dots, M_m\}$ reprezintă o mulțime finită de mașini, unde $m \in \mathbb{N}$;
- $P = \{p_1, p_2, \dots, p_n\}$ reprezintă o mulțime finită de produse, unde $n \in \mathbb{N}$;
- $S = \{s_1, s_2, \dots, s_n\}$ reprezintă o mulțime finită care conține numărul de serii ce trebuie executate din fiecare produs $p_i \in P$, unde $n \in \mathbb{N}$;

- n_i reprezintă numărul de operații necesare pentru finalizarea produsului p_i , $\forall p_i \in P$;
- $\forall p_i \in P$, sunt definite mulțimile finite O_i , LM_i și TP_i astfel:
 - $O_i = \{o_1^i, o_2^i, \dots, o_{n_i}^i\}$ reprezintă un set ordonat de operații necesar execuției produsului p_i ; operațiile sunt definite de o relație binară $o_j^i \rightarrow o_{j+1}^i$ reprezentând restricțiile de precedență între operațiile o_j^i și o_{j+1}^i ;
 - $LM_i = \{m_1^i, m_2^i, \dots, m_{n_i}^i\}$ reprezintă o listă ordonată de mașini, fiecărei operații o_j^i îi corespunde o mașină m_j^i ;
 - $TP_i = \{tp_1^i, tp_2^i, \dots, tp_{n_i}^i\}$ reprezintă o listă a timpilor de execuție asociați fiecărei operații;
 - $Wt_i = \{wt_1^i, wt_2^i, \dots, wt_{n_i}^i\}$ reprezintă o listă a timpilor de așteptare a operației o_j^i pentru a accesa mașina corespunzătoare m_j^i .

O soluție a problemei este reprezentată de un plan $\pi = (p_i, s_j, \pi_{ij})$, unde:

- $\pi_{ij} = (O_i, LM_i, TA_i, TE_i), \forall i \in \overline{1:n}$;
- $TA_i = \{ta_1^i, ta_2^i, \dots, ta_{n_i}^i\}$ reprezintă o listă a timpilor de accesare ai mașinilor;
- $TE_i = \{te_1^i, te_2^i, \dots, te_{n_i}^i\}$ reprezintă o listă a timpilor de eliberare ai mașinilor.

Luând în considerare notațiile de mai sus, unul dintre obiective este acela de a minimiza timpul de start t_j^i al fiecărei operații $o_j^i \in O_i$.

$$\forall o_j^i \in O_i, t_j^i \geq 0 \quad (1.1)$$

$$\forall o_{j_1}^i, o_{j_2}^i \in O_i, o_{j_1}^i \rightarrow o_{j_2}^i: t_{j_2}^i \geq t_{j_1}^i + tp_{j_1}^i \quad (1.2)$$

$$\forall o_{j_1}^i, o_{j_2}^i \in O_i, j_1 \neq j_2, m_{j_1}^i = m_{j_2}^i: (t_{j_2}^i \geq t_{j_1}^i + tp_{j_1}^i) \vee (t_{j_1}^i \geq t_{j_2}^i + tp_{j_2}^i) \quad (1.3)$$

Se notează cu NT_i numărul total de operații necesare produsului $p_i \in P$.

Pentru a identifica valoarea timpului total necesar execuției notat cu C_{max} se ia în considerare următoarea relație:

$$\forall o_j^i \in O_i, C_{max} \geq t_{N_j}^i + tp_{N_j}^i \quad (1.4)$$

Obiectivele principale ale problemei de planificare se referă la: minimizarea timpului de start t_j^i al fiecărei operațiuni $o_j^i \in O_i$, minimizarea timpului de așteptare Id_i pentru accesarea fiecărei mașini M_i , precum și a timpului total necesar planificării C_{max} , precum și timpul total de așteptare pentru accesarea mașinilor.

În aceste condiții, obiectivul se poate rezuma la minimizarea funcției $f: \pi \rightarrow [0,1]$ propusă de autoare în ecuația (1.5).

$$f(\pi) = \frac{C_{max} + \sum_{i=1}^n \sum_{j=1}^{S_i} wt_{ij}}{10^3 \sum_{i=1}^n (S_i * p_i)} \quad (1.5)$$

unde 10^3 este un coeficient de corecție determinat experimental pentru a încadra valoarea funcției în intervalul considerat $[0,1]$.

2 TEHNICI DE INTELIGENȚĂ ARTIFICIALĂ COMPUTAȚIONALĂ PENTRU PLANIFICAREA AUTOMATĂ A PRODUCȚIEI ÎN LINII FLEXIBILE DE FABRICAȚIE

Inteligența artificială computațională (IAC) [73] reprezintă un domeniu interdisciplinar extrem de complex, iar principalele tehnici specifice sunt Rețele Neuronale Artificiale, Fuzzy Logic (Logica Fuzzy) și Algoritmi Evolutivi [62], [42], [72].

Algoritmii Evolutivi reprezintă un subdomeniu al IAC ce utilizează un proces iterativ de creștere și dezvoltare a populației de soluții, ce este mai apoi procesată și utilizată pentru o căutare aleatoare a unei soluții optime [40].

Există o serie de algoritmi de optimizare metaeuristici specifici acestor tehnici evolutive, cum ar fi:

- Algoritmi Genetici;
- Memetic Algorithms (Algoritmi memetici);
- Swarm intelligence (Algoritmi de inteligență a roiurilor) [161]:
 - Particle Swarm Optimization (Optimizarea pe baza roiurilor de particule);

- Ant Colony Optimization (Optimizarea cu colonii de furnici);
- Artificial Bee Colony Optimization (Optimizarea cu roiuri de albine);
- Artificial Imune System (Algoritmul Sistemului Imunitar Artificial).

2.1 STADIUL ACTUAL AL CERCETĂRILOR PRIVIND APLICAREA ALGORITMILOR IA PENTRU PROBLEMA JSS

Atenția cercetătorilor s-a concentrat în principal asupra minimizării timpului total asociat procesului de producție, dar și asupra maximizării gradului de utilizare a mașinilor și a minimizării timpilor de așteptare. O sinteză a studiului privind aplicarea algoritmilor specifici IA în rezolvarea problemei JSS în literatura de specialitate este inclusă în **Tabelul 2.13**.

Tabelul 2.13 Sinteza studiului privind aplicarea algoritmilor specifici IA în rezolvarea problemei JSS în literatura de specialitate

Algoritmul IA	Referință bibliografică
PSO	[124], [110], [109], [20], [165], [80], [99], [124]
AG	[168], [98], [153], [82], [8]
ACO	[47], [158], [71], [95], [111]
SA	[162], [94], [24], [31]
ABC	[28], [166]
SE	[88], [163], [28], [59], [25], [102]
SMA	[78], [32], [75], [11]

2.2 METODE HIBRIDE BAZATE PE IA

Una dintre cele mai noi direcții de cercetare se referă la adoptarea unor tehnici hibride, care să încerce să îmbine avantajele unor metode diferite în scopul determinării unor soluții optime pentru problema propusă. O sinteză a studiului privind tehnicile hibride bazate pe IA pentru rezolvarea problemei JSS disponibile în literatura de specialitate este inclusă în **Tabelul 2.14**.

Tabelul 2.14 Sinteza studiului realizat privind metodele hibride bazate pe IA

Algoritmii ce stau la baza metodei hibride propuse	Referință bibliografică
PSO, TS	[123]
ABC, PA	[77]
AG, SA	[141]
ACO, TS	[44]
PSO, SA	[51], [65], [157], [128], [127]
PSO, AG, Schema de alocare Kacem	[142]
AG, Local Search	[117], [56]
AG, PSO	[79]
PSO,AIS (Artificial Immune System)	[62], [114]
PSO, Algoritmul G&T	[129]
ABC, Logica Fuzzy	[49]
AG, Logica Fuzzy, Sisteme expert	[145]
PSO, MBO (Micro-Migrating Birds Optimizer)	[50]
ABC, PSO, Strategia Greedy	[70]

3 CERCETĂRI EXPERIMENTALE PRIVIND IDENTIFICAREA UNEI METODES HIBRIDE EFICIENTE DE REZOLVARE A PROBLEMEI DE PLANIFICARE JSS

În acest capitol este prezentat un studiu asupra aplicării tehnicilor de Inteligență Artificială Compuțională în domeniul JSS, iar rezultatele obținute atât prin testarea metodelor propuse pentru o serie de benchmark-uri clasice, dar și pentru un set de date de test fictive sunt comparate pe baza unui set de criterii pentru a evidenția algoritmul cu cele mai bune rezultate.

Metoda hibridă va fi bazată pe îmbinarea a două tehnici specifice inteligenței artificiale computaționale (alese conform studiului comparativ), iar pseudocodul acestora va fi prezentat în detaliu în acest capitol. Validarea metodei se va face pe baza unui set de benchmark-uri specifice JSS consacrate în literatura de specialitate.

ANALIZA COMPARATIVĂ A TEHNICILOR DE INTELIGENȚĂ ARTIFICIALĂ COMPUTAȚIONALĂ PENTRU REZOLVAREA PROBLEMEI JSS

În această secțiune a lucrării este prezentat un studiu al sistemelor de planificare bazate pe tehnici de inteligență computațională dezvoltate. Algoritmii utilizați pentru dezvoltarea acestor sisteme sunt PSO, ACO și AG, iar testele au fost făcute atât pentru un set de 15 benchmark-uri clasice, cât și pentru un set de date fictive.

Scopul acestui studiu este acela de a scoate în evidență algoritmi care obțin cele mai bune rezultate în condițiile testate, pentru ca mai apoi aceștia să stea la baza algoritmului hibrid ce va fi propus.

Studiu de caz 1 – Set de benchmark-uri

Algoritmii specifici inteligenței artificiale computaționale au fost testați în primul rând pentru un set de date de test recunoscute în literatura de specialitate (prezentate pe larg în Anexa 4).

Rezultatele obținute în urma rulării algoritmilor pentru parametrii specifici prezentați mai sus sunt sintetizate în **Tabelul 3.5**. Este de remarcat faptul că în 13 cazuri valoarea cea mai mică este obținută în urma algoritmului PSO, iar în 2 cazuri în urma algoritmului ACO, rezultatele AG fiind cele mai slabe în condițiile testate.

Pentru benchmark-urile ABZ5, FT6/MT06, FT10/MT10 și LA02 algoritmi ACO și PSO implementați reușesc să obțină rezultate încadrate în intervalul definit de limita superioară și limita inferioară acceptate. Referitor la seturile de parametri specifici, se poate observa faptul că valorile optime obținute în urma testării PSO sunt corespunzătoare unei populații cu un număr mai mare de particule și unei valori medii pentru parametrul cognitiv și cel social. De asemenea, în ceea ce privește ACO, seturile de parametri specifici corespunzător celor mai bune rezultate sunt caracterizate de valori mai mari pentru numărul de furnici artificiale, o valoare medie a ratei de evaporare a feromonului, precum și o valoare a influenței urmei de feromon mai mică.

Tabelul 3.5 Rezultate experimentale

Denumirea benchmark-ului	Limita inferioară acceptată	Limita superioară acceptată	Valoarea optimă recunoscută	Valoarea optimă obținută cu AG	Valoarea optimă obținută cu ACO	Valoarea optimă obținută cu PSO	Setul de parametrii specifici utilizați
ABZ5	868	1370	1234	1360	1320	1320	PSO_PS15 ACO_PS12
ABZ6	742	1071	943	1102	1095	1086	PSO_PS8
FT6 / MT06	46	68	55	72	65	68	ACO_PS8
FT10 / MT10	655	1262	930	1360	1265	1252	PSO_PS13
LA01	666	830	666	831	823	794	PSO_PS14
LA02	635	844	655	935	842	876	ACO_PS12
LA03	588	773	597	793	865	790	PSO_PS13
LA04	537	839	590	1025	1108	1005	PSO_PS14
LA19	685	1046	842	1581	1494	1383	PSO_PS14
LA20	756	1210	902	1782	1705	1666	PSO_PS15
ORB1	695	1456	1059	1830	1841	1727	PSO_PS15
ORB2	671	1157	888	1292	1386	1255	PSO_PS14
ORB3	648	1297	1005	1465	1532	1433	PSO_PS15
ORB4	759	1356	1005	1888	1865	1755	PSO_PS15
ORB5	630	1116	887	1400	1369	1304	PSO_PS14

Astfel în cazul algoritmului ACO, pentru setul de parametrii specifici ACO_PS8, se obțin cele mai bune rezultate în 6.66% din cazuri, iar pentru ACO_PS12 în 13.33% din cazuri. În cazul algoritmului PSO, pentru setul de parametrii specifici PSO_PS8 se obțin cele mai bune rezultate în 6.66% din cazuri, pentru PSO_PS13 în 13.33% din cazuri, pentru PSO_PS14 în 33.33% din cazuri, iar pentru PSO_PS15 în 26.66% din cazuri. În total cu ajutorul algoritmului PSO se obțin cele mai bune rezultate în 86,66% din cazuri, cu ACO în 20% din cazuri, iar algoritmul AG nu obține în nici unul dintre cazuri cel mai bun rezultat.

Studiu de caz 2 – Set de date de test fictive

Algoritmii ACO, PSO și AG au fost testați pentru trei scenarii fictive diferite corespunzătoare structurii sistemului de programare a producției intermitente (JSS) așa

cum este prezentat în **Tabelul 3.7**. Scopul acestor teste este acela de a identifica modul în care aceștia se comportă în situații diferite de producție.

De exemplu, pentru scenariul JSSP_1 numărul de operații necesare pentru produsul de tip P_1 este 2, iar aceste operații trebuie executate pe mașinile M1 și M3, cu timpii de procesare 2 și respectiv 2; produsul P_2 necesită un număr de 2 operații ce trebuie executate pe mașinile M3 și M5, cu timpii de procesare 3 și respectiv 2 și așa mai departe.

Tabelul 3.7 Scenariile JSS

Scenariul JSS	Tipul produsului	Numărul de operații	Secvența de mașini / Timpul de procesare (minute)
JSSP_1	P_1	2	M1/2, M3/2
	P_2	2	M3/3, M5/2
	P_3	3	M2/4, M4/3, M1/4
	P_4	2	M2/3, M5/3
	P_5	4	M3/3, M1/2, M4/3, M5/2
JSSP_2	P_1	3	M1/2, M3/2, M2/3
	P_2	2	M4/3, M1/2
	P_3	3	M2/2, M3/3, M5/4
	P_4	2	M2/3, M5/4
	P_5	4	M3/3, M1/2, M4/5, M5/2
JSSP_3	P_1	2	M1/2, M3/2
	P_2	2	M3/2, M4/3
	P_3	3	M2/3, M5/5, M1/3
	P_4	2	M1/3, M3/4
	P_5	5	M3/2, M1/2, M4/3, M5/2, M2/2

Fiecare scenariu JSS a fost testat pentru 5 seturi de date de intrare, reprezentând cerințe de producție diferite pentru numărul de serii necesar din fiecare produs în parte, așa cum este prezentat în **Tabelul 3.8**.

Algoritmii implementați au fost comparați luând în considerare următoarele criterii: calitatea soluției (reprezentată prin valoarea funcției f), timpul total necesar execuției (C_{max}), timpul de rulare și numărul de soluții obținute.

Tabelul 3.9 prezintă valorile obținute în urma rulării celor 5 algoritmi implementați referitoare la funcția f calculate pentru soluția obținută.

Tabelul 3.8 Cerințe diferite de producție

Numărul setului de date	Numărul de serii/produs					Numărul de simulări
	P1	P2	P3	P4	P5	
DS1	20	10	20	20	10	10
DS2	50	0	0	50	60	10
DS3	30	40	20	0	50	10
DS4	40	40	40	40	30	10
DS5	50	60	50	60	60	10

Având în vedere faptul că valoarea funcției f trebuie să fie minimizată, conform modelului matematic din secțiunea 1.5, se poate observa că soluțiile obținute prin PSO au cea mai mică valoare a funcției f în 73,33% din cazuri, iar soluțiile AG obțin această valoare minimă a lui f în 53,33% din cazuri. Trebuie notat faptul că aceste valori mici ale soluțiilor obținute prin AG sunt corespunzătoare cazurilor cu cea mai mare complexitate referitoare la numărul necesar de serii de produse. Soluțiile obținute prin ACO sunt corespunzătoare unei valori minime a funcției f în 26,66% din cazuri. Algoritmii FCFS și RR duc la soluții slabe calitativ comparativ cu ceilalți trei algoritmi.

Din **Tabelul 3.10** se pot extrage următoarele informații: soluțiile obținute prin PSO au cea mai mică valoare pentru C_{max} în 73,33% din cazuri, iar soluțiile AG obțin această valoare minimă în 53,33% din cazuri. Trebuie notat faptul că aceste valori minime ale timpului total de execuție obținute prin PSO sunt corespunzătoare cazurilor cu cea mai mare complexitate referitoare la numărul necesar de serii de produse. Soluțiile obținute prin ACO sunt corespunzătoare unei valori minime a lui C_{max} în 26,66% din cazuri. Algoritmii FCFS și RR conduc la soluții slabe calitativ comparativ cu ceilalți trei algoritmi.

Tabelul 3.9 Rezultate comparative referitoare la valoarea funcției f

Scenariul JSS	Numărul setului de date	Funcția fitness f				
		AG	ACO	PSO	FCFS	RR
JSSP_1	DS1	0.035788	0.036936	0.035316	0.048264	0.054093
	DS2	0.086084	0.080548	0.085578	0.105192	0.111092
	DS3	0.092975	0.093905	0.092975	0.128344	0.126590
	DS4	0.096937	0.096440	0.096440	0.157389	0.158933
	DS5	0.190678	0.191901	0.190678	0.300639	0.324517

JSSP_2	DS1	0.048900	0.049192	0.049192	0.056205	0.055382
	DS2	0.112392	0.109550	0.112392	0.129778	0.133221
	DS3	0.105751	0.108684	0.105751	0.105758	0.112300
	DS4	0.126942	0.122946	0.126942	0.144451	0.152768
	DS5	0.229766	0.233702	0.229766	0.263935	0.296572
JSSP_3	DS1	0.045382	0.046006	0.045382	0.054060	0.069871
	DS2	0.125772	0.127618	0.125103	0.169692	0.180128
	DS3	0.085486	0.091442	0.085486	0.127414	0.138431
	DS4	0.035788	0.036936	0.035316	0.048264	0.054093
	DS5	0.086084	0.080548	0.085578	0.105192	0.111092

Tabelul 3.10 Rezultate comparative referitoare la valoarea lui C_{max}

Scenariul JSS	Numărul setului de date	C_{max}				
		AG	ACO	PSO	FCFS	RR
JSSP_1	DS1	161	162	157	210	211
	DS2	326	306	326	387	392
	DS3	356	359	356	475	491
	DS4	345	344	344	540	553
	DS5	540	545	540	800	792
JSSP_2	DS1	209	213	213	245	269
	DS2	390	382	390	455	470
	DS3	388	398	388	394	428
	DS4	433	419	433	505	503
	DS5	599	613	599	715	747
JSSP_3	DS1	201	198	201	235	250
	DS2	448	457	446	589	597
	DS3	321	343	321	475	468
	DS4	425	453	423	495	547
	DS5	648	656	648	735	831

METODA HIBRIDĂ HIBRIPSO-SA – PREZENTARE ALGORITM

Conform rezultatelor prezentate în secțiunea 3.1. a acestei lucrări, metoda cea mai potrivită pentru rezolvarea JSS este PSO deoarece aceasta reușește să obțină rezultate

superioare față de celelalte metode luate în considerare. Dezavantajul principal al acestei metode este legat de faptul că se poate bloca în căutare în punctele de optim local.

În continuare este prezentată metoda hibridă de planificare HibriPSO-SA bazată pe combinarea PSO cu SA. Această metodă încearcă să îmbine avantajele celor două tehnici de optimizare adaptate pentru problema JSS, prin combinarea strategiei de căutare locală cu cea de căutare globală.

Considerând algoritmul PSO și algoritmul SA prezentate în secțiunea 2.2 și, respectiv, 2.4, *elementele de noutate* aduse de către metoda hibridă HibriPSO-SA propusă sunt următoarele:

- Populația inițială de particule se obține astfel: pentru particula generată aleator este aplicat algoritmul SA, și abia apoi este inclusă în populația inițială pentru PSO.

Pentru $i=1$ la N_p execută

Generează ($Part_i$)

$Part_i' = SA(Part_i)$

$Part_i = Part_i'$

- Pentru fiecare pas al algoritmului PSO, se selectează aleator o serie de particule cărora li se aplică algoritmul SA. Aceste particule modificate vor fi integrate în populația curentă doar în cazul în care valoarea funcției fitness a acesteia este mai mică decât β , unde β este pragul de admisibilitate al unei particule și este egal cu valoarea medie a tuturor funcțiilor fitness corespunzătoare indivizilor din populația curentă. Din cercetările experimentale realizate s-a ajuns la concluzia că pentru îmbunătățirea vitezei de rulare este utilă aplicarea unei funcții probabilistice ce presupune determinarea aleatoare a unei valori β' dintr-un interval $[a', b']$, unde a' și b' au fost determinate experimental și luarea în considerare a particulei modificate pentru calculul funcției fitness aferente și adăugarea la populația curentă doar în cazul în care $\beta' \geq (a' + b')/2$.

Număr_particule_pentru_SA=random(1,Np/2)

Pentru i=1 la Număr_particule_pentru_SA execută

Particule_selectate_pentru_SAi=Selectează_aleator_particula()

Pentru i=1 la Număr_particule_pentru_SA execută

$\beta' = \text{random}(1, Np/2)$

Dacă $\beta' \geq (a' + b')/2$ atunci

Partaux=SA(Particule_selectate_pentru_SAi)

Calculează_fitness(Partaux)

Dacă $\text{fitness}(Partaux) \leq \beta$ atunci

Include_in_populatia_curentă(Partaux)

- Fiecare particulă este verificată pentru a identifica eventualele discrepanțe ce pot apărea în cadrul planificării propuse, iar în cazul în care acestea apar, vor fi efectuate modificările necesare pentru a obține o soluție fezabilă (Codificarea în limbajul C a acestei funcții este prezentată în Anexa 6).

Pentru i=1 la Np execută

Elimina_timpi_de_asteptare_nejustificati (Parti)

- Utilizarea unei funcții obiectiv complexe (denumită funcție fitness) pentru o evaluare cât mai corectă a calității soluțiilor propuse (Codificarea în limbajul C este prezentată în Anexa 7). Formula acestei funcții este construită astfel încât să ofere o imagine cât mai corectă asupra soluției evaluate, luând în considerare pe lângă valoarea timpului total asociat planului și valorile asociate timpilor de așteptare.

Pentru i=1 la Np execută

Calculează_fitness(Parti)

Astfel datele de intrare pentru algoritmul HibridPSO-SA vor fi compuse din datele de intrare specifice problemei JSSP descrisă prin modelul matematic din secțiunea 1.5 și datele de intrare celor doi algoritmi ce stau la baza metodei hibride: SA și PSO. Codificarea în limbajul C este prezentată în Anexa 6.

Reprezentare soluție

Soluția candidat reprezentată printr-o particulă *Part* ce face parte din populația *Hibri_Populatie* va fi structurată astfel:

- $s_i^k \in S$ reprezintă numărul de serii executate din produsul $p_i \in P$ până la pasul k ;
- $\pi = (p_i, s_i^k, \pi_{ij})$ reprezintă planul de fabricație, unde:
 - $\pi_{ij} = (O_i, LM_i, TA_i, TE_i), \forall i \in \overline{1:n}$;
 - $TA_i = \{ta_1^i, ta_2^i, \dots, ta_{n_i}^i\}$ reprezintă o listă a timpilor de accesare ai mașinilor;
 - $TE_i = \{te_1^i, te_2^i, \dots, te_{n_i}^i\}$ reprezintă o listă a timpilor de eliberare ai mașinilor.
- $pBest$ reprezintă cea mai bună valoare a funcției fitness până în momentul respectiv;
- $\pi' = (p_i', s_i^{k'}, \pi_{ij}')$ reprezintă planul de fabricație cu care particula a atins valoarea $pBest$;
- $V_j = \{v_1^j, v_2^j, \dots, v_{n_i}^j\}$ reprezintă viteza particulei $j, j \in \overline{1:Np}$.

Calitatea soluției obținute de fiecare particulă este calculată cu ajutorul unei funcții fitness $f_{HibriPSO_SA}: Hibri_Populatie \rightarrow [0,1]$ ce urmărește minimizarea timpului total necesar execuției, precum și minimizarea timpilor de așteptare pentru accesarea mașinilor. Forma acestei funcții propusă de autoare este prezentată în ecuația (3.1).

$$f_{HibriPSO_SA} = \frac{C_{\max} + \sum_{i=1}^n \sum_{j=1}^{s_i} wt_{ij}}{10^3 \sum_{i=1}^n (s_i * p_i)} \quad (3.1)$$

unde 10^3 este un coeficient de corecție determinat experimental pentru a încadra valoarea funcției în intervalul considerat $[0,1]$.

Criteriul de finalizare a algoritmului este reprezentat de atingerea numărului total de iterații. Soluția finală va fi reprezentată de individul cu cea mai mică funcție fitness obținut în urma algoritmului HibriPSO-SA.

Algoritm 3.1 - Algoritm Hibrid PSO-SA pentru JSS

Date de intrare specifice JSS:

- ✓ m - numărul de mașini disponibile;
- ✓ n - numărul de produse;
- ✓ $LM = \{M_1, M_2, \dots, M_m\}$ - lista mașinilor din sistemul de fabricație;
- ✓ $P = \{p_1, p_2, \dots, p_n\}$ - lista produselor ce vor fi lansate în execuție;
- ✓ $s_i \in S$ - numărul de serii ce trebuie executate pentru produsul p_i , $\forall p_i \in P$;
- ✓ $O_i = \{o_1^i, o_2^i, \dots, o_{n_i}^i\}$ - un set ordonat de operații necesar execuției produsului p_i ;
- ✓ $LM_i = \{m_1^i, m_2^i, \dots, m_{n_i}^i\}$ - lista ordonată de mașini, fiecărei operații o_j^i îi corespunde o mașină m_j^i , $\forall p_i \in P$;
- ✓ $TP_i = \{tp_1^i, tp_2^i, \dots, tp_{n_i}^i\}$ - listă a timpilor de execuție asociați fiecărei operații, $\forall p_i \in P$;

Date de intrare specifice SA:

- ✓ T_0 - temperatura inițială;
- ✓ α - rata de răcire;

Date de intrare specifice PSO:

- ✓ N_p - numărul de particule disponibile;
- ✓ $\phi_1 \in (0,1)$ - parametrul cognitiv;
- ✓ $\phi_2 \in (0,1)$ - parametrul social;
- ✓ I_{\max} - numărul maxim de iterații;

Date de ieșire:

- ✓ $f(gBest)$
- ✓ $\pi_{gBest} = (p_i, s_i^k, \pi_{ij})$ - planul de fabricație corespunzător celei mai bune particule
- ✓ C_{\max}

Început

1. Citește datele de intrare
2. Pentru $i=1, N_p$ execută
 - 2.1. Inițializează mulțimea $Produse_Disponibile=P$
 - 2.2. $k=1$
 - 2.3. **Cât timp** $Produse_Disponibile \neq \emptyset$ execută
 - 2.3.1. Alege aleatoriu produsul p_i din mulțimea $Produse_Disponibile$
 - 2.3.2. $Part.produs[k]=p_i$
 - 2.3.3. $Part.serie[k]=s_i$

- 2.3.4. $s_i = s_i - 1$
- 2.3.5. **Dacă** $s_i = 0$ **atunci**
 $Produse_Disponibile = Produse_Disponibile \setminus \{p_i\}$
- 2.4. **Pentru** $k=1, (s_1 + \dots + s_n)$ **execută**
- 2.4.1. **Pentru** $l=1, n_{Part.prod[k]}$ **execută**
- 2.4.1.1. Inițializează $ta_l^{Part.prod[k]}$
- 2.4.1.2. Calculează $te_l^{Part.prod[k]}$
- 2.5. Calculează $f(Part[i])$
- 2.6. Inițializează $T=T_0$
- 2.7. Starea curentă $sBest = Part[i]$
- 2.8. $f(sBest) = f(Part[i])$
- 2.9. **Cât timp** ($T < T_{min}$) **execută**
- 2.9.1. Inițializează mulțimea $Produse_Disponibile = P$
- 2.9.2. $k=1$
- 2.9.3. **Cât timp** $Produse_Disponibile \neq \emptyset$ **execută**
- 2.9.3.1. Alege aleatoriu produsul p_i din mulțimea $Produse_Disponibile$
- 2.9.3.2. $Part^*.produs[k] = p_i$
- 2.9.3.3. $Part^*.serie[k] = s_i$
- 2.9.3.4. $s_i = s_i - 1$
- 2.9.3.5. **Dacă** $s_i = 0$ **atunci**
 $Produse_Disponibile = Produse_Disponibile \setminus \{p_i\}$
- 2.9.4. **Pentru** $k=1, (s_1 + \dots + s_n)$ **execută**
- 2.9.4.1. **Pentru** $l=1, n_{Part^*.prod[k]}$ **execută**
- 2.9.4.1.1. Inițializează $ta_l^{Part^*.prod[k]}$
- 2.9.4.1.2. Calculează $te_l^{Part^*.prod[k]}$
- 2.9.5. Calculează $f(Part^*)$
- 2.9.6. **Dacă** $f(Part^*) < f(sBest)$ **atunci**
- 2.9.6.1. $f(sBest) = f(Part^*)$
- 2.9.6.2. **Pentru** $k=1, (s_1 + \dots + s_n)$ **execută**
- 2.9.6.2.1. **Pentru** $l=1, n_{Part^*.prod[k]}$ **execută**
- 2.9.6.2.1.1. $ta_l^{sBest} = ta_l^{Part^*.prod[k]}$
- 2.9.6.2.1.2. $te_l^{sBest} = te_l^{Part^*.prod[k]}$
- 2.9.7. Ajustează temperatura $T = T * \alpha$
- 2.9.8. $contor_pas = contor_pas + 1$
- 2.10. Inițializează particula P_i cu soluția obținută $sBest$
3. $pas = 0$
4. **Cât timp** ($pas < I_{max}$) **execută**

- 4.1. Inițializează aleator $Np^* \in (1, Np/2)$
- 4.2. Calculează β
- 4.3. **Pentru** $ii=1, Np^*$ **execută**
 - 4.3.1. Alege aleator o particulă PP^* nemodificată la acest pas din populația curentă Hibri_Populatie
 - 4.3.2. Calculează $f(PP^*)$
 - 4.3.3. Inițializează $T=T_0$
 - 4.3.4. Starea curentă $sBest = PP^*$
 - 4.3.5. $f(sBest) = f(PP^*)$
 - 4.3.6. **Cât timp** ($T < T_{min}$) **execută**
 - 4.3.6.1. Inițializează mulțimea $Produse_Disponibile = P$
 - 4.3.6.2. $k=1$
 - 4.3.6.3. **Cât timp** $Produse_Disponibile \neq \emptyset$
 - 4.3.6.3.1. Alege aleator produsul p_i din mulțimea $Produse_Disponibile$
 - 4.3.6.3.2. $PP^*.produs[k] = p_i$
 - 4.3.6.3.3. $PP^*.serie[k] = s_i$
 - 4.3.6.3.4. $s_i = s_i - 1$
 - 4.3.6.3.5. Dacă $s_i = 0$ atunci
 $Produse_Disponibile = Produse_Disponibile \setminus \{p_i\}$
 - 4.3.6.3.6. **Pentru** $k=1, (s_1 + \dots + s_n)$ **execută**
 - 4.3.6.3.6.1. **Pentru** $l=1, n_{PP^*.prod[k]}$ **execută**
 - 4.3.6.3.6.1.1. Inițializează
 $ta_l^{PP^*.prod[k]}$
 - 4.3.6.3.6.1.2. Calculează
 $te_l^{PP^*.prod[k]}$
 - 4.3.6.3.4. Calculează $f(PP^*)$
 - 4.3.6.5. **Dacă** $f(PP^*) < \beta$ **atunci**
 - 4.3.6.5.1. **Pentru** $k=1, (s_1 + \dots + s_n)$ **execută**
 - 4.3.6.5.1.1. **Pentru** $l=1, n_{PP^*.prod[k]}$ **execută**
 - 4.3.6.5.1.1.1. $ta_l^{sBest} = ta_l^{PP^*.prod[k]}$
 - 4.3.6.5.1.1.2. $te_l^{sBest} = te_l^{PP^*.prod[k]}$
 - 4.3.6.6. Ajustează temperatura $T = T^* \alpha$
 - 4.3.6.7. $contor_pas = contor_pas + 1$
 - 4.3.7. Înlocuiește particula PP^* cu soluția obținută $sBest$
- 4.4. **Pentru** $i=1, Np$ **execută**
 - 4.4.1. **Dacă** $f(gBest) < f(Part[i])$ **atunci**
 - 4.4.1.1. $f(pBest) = f(Part[i])$
 - 4.4.1.2. **Pentru** $k=1, (s_1 + \dots + s_n)$ **execută**

<p>4.4.1.2.1. Pentru $l=1, n_{part.[i]prod[k]}$ execută</p> <p>4.4.1.2.1.1. $ta_l^{gBest.prod[k]} = ta_l^{Part[i].prod[k]}$</p> <p>4.4.1.2.1.2. $te_l^{gBest.prod[k]} = te_l^{Part[i].prod[k]}$</p> <p>4.5. Pentru $P=1, Np$ execută</p> <p>4.5.1. Pentru $k=1, (s_1+...+s_n)$ execută</p> <p>4.5.1.1. Pentru $l=1, n_{part.prod[k]}$ execută</p> <p>4.5.1.1.1. $v_k^{part} = v_k^{part} + rand(\phi_1) * ta_l^{gBest.prod[k]} + rand(\phi_2) * ta_l^{Part.pBest.prod[k]}$</p> <p>4.5.1.1.2. $ta_l^{Part.prod[k]} = ta_l^{Part.prod[k]} + v_k^{part}$</p> <p>4.5.1.1.3. Calculează $te_l^{Part.prod[k]}$</p> <p>4.5.1.1.4. Liniarizare <i>particulă</i>(Part)</p> <p>4.6. $pas=pas+1$</p> <p>5. Calculează C_{max} pentru $gBest$</p> <p>6. Afișează $gBest$</p> <p>Sfârșit</p>

3.1 VALIDAREA ALGORITMULUI HIBRIPSO-SA

Metoda hibridă a fost validată pe date de test consacrate în literatura de specialitate (descrise în Anexa 4). Rezultatele obținute sunt apoi comparate cu rezultatele deja validate pentru aceste seturi de date de test pentru a evidenția gradul de aplicabilitate a algoritmului hibrid în domeniul problemei propuse.

Au fost luate în considerare 15 benchmark-uri clasice pentru literatura de specialitate, așa cum este prezentat în **Tabelul 3.13**. Având în vedere faptul că metoda hibridă propusă se bazează pe două tehnici specifice inteligenței artificiale computaționale, care evoluează diferit în funcție de valorile parametrilor specifici, testele au fost efectuate pentru o serie de 20 de seturi diferite pentru aceștia, așa cum este prezentat în **Tabelul 3.15**.

Astfel în cazul benchmark-urilor ABZ5, FT6/MT06, FT10/MT10, LA02, LA04, LA19, LA20, ORB1, ORB2, ORB3 și ORB5, valoarea optimă obținută în urma rulării algoritmului HibriPSO-SA este aceeași cu valoarea optimă recunoscută pentru benchmark-ul respectiv. În celelalte cazuri (ABZ6, LA01, LA03 și ORB4) valoarea

obținută este apropiată de valoarea optimă și se încadrează în intervalul acceptat de către literatura de specialitate.

Rezultatele experimentale prezentate în **Tabelul 3.15** subliniază faptul că soluțiile obținute prin aplicarea metodei hibride propuse referitoare la timpul total necesar execuției reușesc să se încadreze în intervalul determinat de limita inferioară și limita superioară acceptate în 100% din cazuri. De asemenea în 73.33% dintre cazurile considerate, algoritmul HibriPSO-SA reușește să obțină un plan de fabricație al cărui timp total necesar execuției este egal cu valoarea optimă recunoscută în literatura de specialitate. În Anexa 4 este prezentată o comparație a rezultatelor obținute prin metoda hibridă atât pentru benchmark-urile luate în considerare, cât și pentru datele de test fictive cu algoritmi specifici IA (ACO, PSO, AG și SA), dar și cu algoritmi clasici (FCFS și RR). Aceste rezultate subliniază faptul că metoda hibridă obține rezultatele cele mai bune în toate cazurile testate.

Tabelul 3.15. Rezultate experimentale

Denumirea benchmark-ului	Limita inferioară acceptată	Limita superioară acceptată	Valoarea optimă recunoscută	Valoarea optimă obținută cu HibriPSO-SA
ABZ5	868	1370	1234	1234
ABZ6	742	1071	943	948
FT6 / MT06	46	68	55	55
FT10 / MT10	655	1262	930	930
LA01	666	830	666	671
LA02	635	844	655	655
LA03	588	773	597	598
LA04	537	839	590	590
LA19	685	1046	842	842
LA20	756	1210	902	902
ORB1	695	1456	1059	1059
ORB2	671	1157	888	888
ORB3	648	1297	1005	1005
ORB4	759	1356	1005	1020
ORB5	630	1116	887	887

În **Tabelul 3.16** sunt prezentate rezultatele experimentale referitoare la timpul de rulare, marcându-se totodată și setul de parametri specifici metodei hibride propuse ce au dus la soluția optimă. Se poate observa faptul că valorile optime obținute în urma

rulării algoritmului HibriPSO-SA sunt obținute în 40% din cazuri pentru setul de date PS18, în 40% din cazuri pentru setul de date PS16, în 13% din cazuri pentru setul de date PS8 și în 5% din cazuri pentru setul de date S7.

În urma analizei rezultatelor obținute se poate concluziona faptul că pentru a obține rezultate cât mai bune, parametrii specifici algoritmului hibrid ar trebui ajustați astfel încât numărul de particule disponibile să fie mare (atât în PS18 cât și în S16 N_p este egal cu 100), iar valorile asociate parametrului cognitiv și parametrului social să fie în jurul jumătății intervalului disponibil (în cazul PS18, valorile considerate pentru ϕ_1 și ϕ_2 sunt egale cu 0.6, iar în cazul PS16, valoarea considerată pentru ϕ_1 este de 0.16 și pentru ϕ_2 0.5).

Tabelul 3.16. Rezultate experimentale

Denumirea benchmark-ului	Valoarea optimă pentru HibriPSO-SA a lui C_{max}	Timpul de rulare (milisecunde)	Denumirea setului de parametrii specifici HibriPSO-SA
ABZ5	1234	87458	PS18
ABZ6	948	94658	PS16
FT6 / MT06	55	9296	PS16
FT10 / MT10	930	145301	PS18
LA01	671	10896	PS16
LA02	655	10122	PS18
LA03	598	11765	PS7
LA04	590	11021	PS8
LA19	842	10270	PS16
LA20	902	11891	PS18
ORB1	1059	138536	PS18
ORB2	888	124744	PS8
ORB3	1005	133435	PS18
ORB4	1020	137829	PS16
ORB5	887	125469	PS16

Rezultatele experimentale obținute pe baza datelor de test evidențiază faptul că această metodă conduce la obținerea unor rezultate de o calitate bună, ce reușesc să se încadreze în limitele recomandate, rezultatele susținând ideea că metoda propusă este una potrivită pentru optimizarea planificării producției.

4 DEZVOLTAREA SISTEMULUI INTELIGENT DE PLANIFICARE AUTOMATĂ HIBRIPLANJSS BAZAT PE METODA HIBRIDĂ PROPUȘĂ

În acest capitol sunt prezentate cercetările experimentale privind dezvoltarea unui sistem inteligent de planificare a producției (HibriPlanJSS) bazat pe metoda hibridă HibriPSO-SA descrisă în capitolul 3. Această metodă de planificare este bazată pe cele două tehnici selectate (PSO și SA) și încearcă să îmbine avantajele acestora pentru problema JSS, prin combinarea strategiei de căutare locală cu cea de căutare globală.

4.1 METODOLOGIA UTILIZATĂ PENTRU DEZVOLTAREA SISTEMULUI DE PLANIFICARE

Dezvoltarea sistemului inteligent pentru planificarea automată HibriPlanJSS în linii flexibile de fabricație s-a realizat conform următoarei metodologii:

1. *Descrierea modelului matematic asociat problemei de planificare.* Constă în elaborarea unui model matematic pentru problema propusă care să descrie în detaliu toate caracteristicile și particularitățile acesteia (Secțiunea 1.5).
2. *Analiza comparativă a eficienței algoritmilor specifici inteligenței artificiale pentru problema propusă.* Această etapă se referă la testarea unor algoritmi specifici inteligenței artificiale (Optimizarea prin modelul coloniilor de furnici – Ant Colony Optimization, Optimizarea prin modelul ansamblurilor de particule – Particle Swarm Optimization, Algoritmi Genetici, Călirea simulată – Simulated annealing) în contextul problemei JSS. Rezultatele comparației stau la baza alegerii algoritmilor ce vor fi parte componentă a metodei hibride (Secțiunea 3.1).
3. Elaborarea și implementarea metodei hibride de planificare pe baza rezultatelor studiului comparativ. Constă în dezvoltarea algoritmului hibrid pentru planificare bazat pe PSO și SA (Secțiunea 3.2).

4. *Modelarea sistemului utilizând rețele Petri.* Constă în utilizarea formalismului rețelelor Petri pentru modelarea acțiunilor ce presupun utilizarea în comun a resurselor disponibile în scopul de a realiza succesiunea de operații necesare (Secțiunea 4.3).
5. *Validarea și testarea metodei hibride de planificare.* Algoritmul hibrid este supus etapei de validare ce constă în evaluarea acestuia pe baza unor seturi de date clasice de testare, dar și pe baza unui set de date fictive (Secțiunea 3.3). Apoi algoritmul este testat pe baza unor seturi de date de intrare caracteristice unei linii flexibile de fabricație reale pentru producția de margarină disponibile în cadrul Unilever România (Secțiunea 4.5).
6. *Implementarea sistemului HibriPlanJSS.* Constă în implementarea sistemului de planificare HibriPlanJSS ce are la bază metoda hibridă propusă utilizând platforma QtCreator 2.6.2 (Secțiunea 4.4).

4.2 STRUCTURA SISTEMULUI INTELIGENT DE PLANIFICARE

Structura generală a sistemului inteligent de planificare HibriPlanJSS bazat pe metoda hibridă Hibri-PSO-SA este prezentată în **Figura 4.1** și reprezintă o adaptare a structurii prezentate în **Figura 1.3**.

Intrările sistemului HibriPlanJSS sunt reprezentate de parametrii specifici algoritmului hibrid precum și de structura funcției fitness implementate care are rolul de a evalua calitatea fiecărei soluții candidat.

Ieșirile sistemului HibriPlanJSS sunt reprezentate de o serie de specificații detaliate de planificare a operațiilor în cadrul procesului de producție, cât și de valoarea timpului total necesar execuției planului C_{max} .

Perturbațiile sunt reprezentate de modificările ce pot apărea în starea generală a sistemului de producție (modificările apărute în comenzi, eventualele lipsuri ale materiei prime sau defecțiunile mașinilor-unelte).

Procesul de producție primește ca intrări specificațiile de planificare a operațiilor corespunzătoare planului oferit de sistemul de planificare HibriPlanJSS, împreună cu valoarea timpului total necesar execuției.

Ieșirile procesului de producție sunt reprezentate de produsele finite obținute în urma respectării planului de fabricație (P_1, P_2, \dots, P_n)

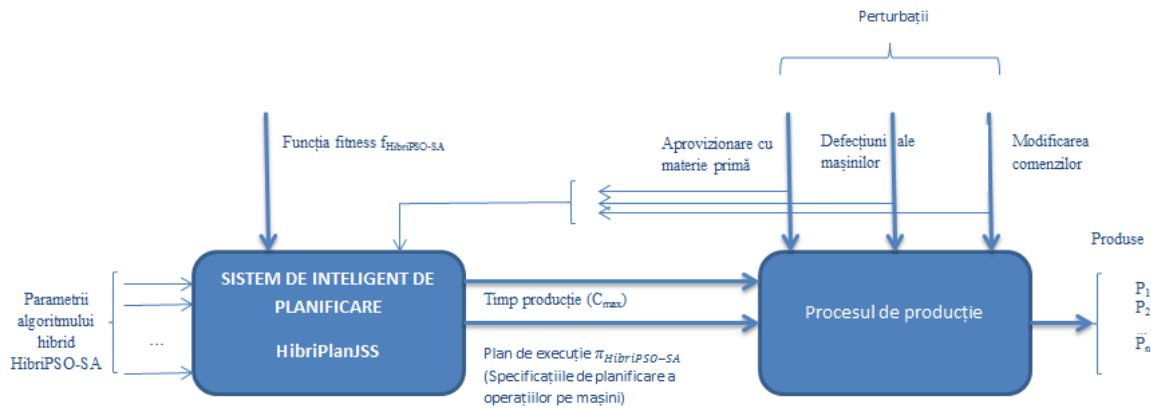


Figura 4.1 Structura generală a sistemului inteligent de planificare HibriPlanJSS

4.3 MODELAREA CU REȚELE PETRI

Studiul sistemelor cu evenimente discrete a avut un impact puternic asupra anumitor arii din inginerie cum ar fi sistemele de fabricație, sistemele de transport, sistemele de operare, sistemele de comunicații [104].

Modelarea rețelei Petri în PIPE v.4.3.0

Pentru modelarea rețelei Petri corespunzătoare sistemului de fabricație de la Unilever România a fost folosit instrumentul software PIPE (Platform Independent Petri net Editor [18]), versiunea v4.3.0.

Graful de accesibilitate

Graful de accesibilitate corespunzător rețelei Petri ce descrie funcționarea liniei flexibile de fabricație din cadrul Unilever este prezentat în **Figura 4.12**.

În urma modelării cu rețele Petri a sistemului flexibil de fabricație specializat pentru producția de margarină disponibil la Unilever România se poate remarca faptul că modelul rețelei a fost validat, iar gradul de accesibilitate este finit. În aceste condiții se poate spune că modelul propus este unul viabil și funcționează într-o manieră corectă.

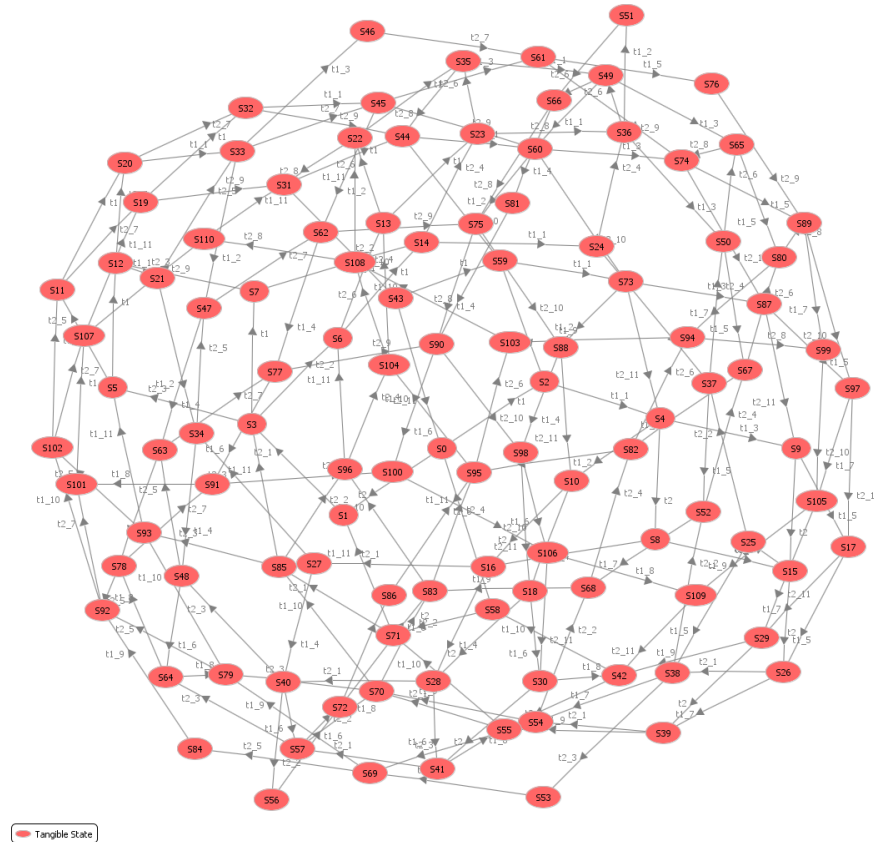


Figura 4.12 Graful de accesibilitate

4.4 IMPLEMENTAREA SISTEMULUI INTELIGENT DE PLANIFICARE AUTOMATĂ A PRODUCȚIEI

4.4.1 Studiu de caz – Unilever România

Pentru testarea algoritmului HibriPSO-SA pe o linie de fabricație reală, a fost efectuat un studiu de caz în cadrul Unilever România, pe zona de producție de margarină.

Unilever reprezintă una dintre cele mai mari companii din România de pe piața bunurilor de larg consum (produse alimentare, produse de îngrijire a locuinței și produse

de îngrijire personală), iar cea mai mare parte dintre produsele acestei companii sunt realizate pe platforma industrială din Ploiești.

În **Figura 4.14** este prezentată structura generală a celor 4 linii flexibile de fabricație specializate pentru producția de margarină din cadrul Unilever România.

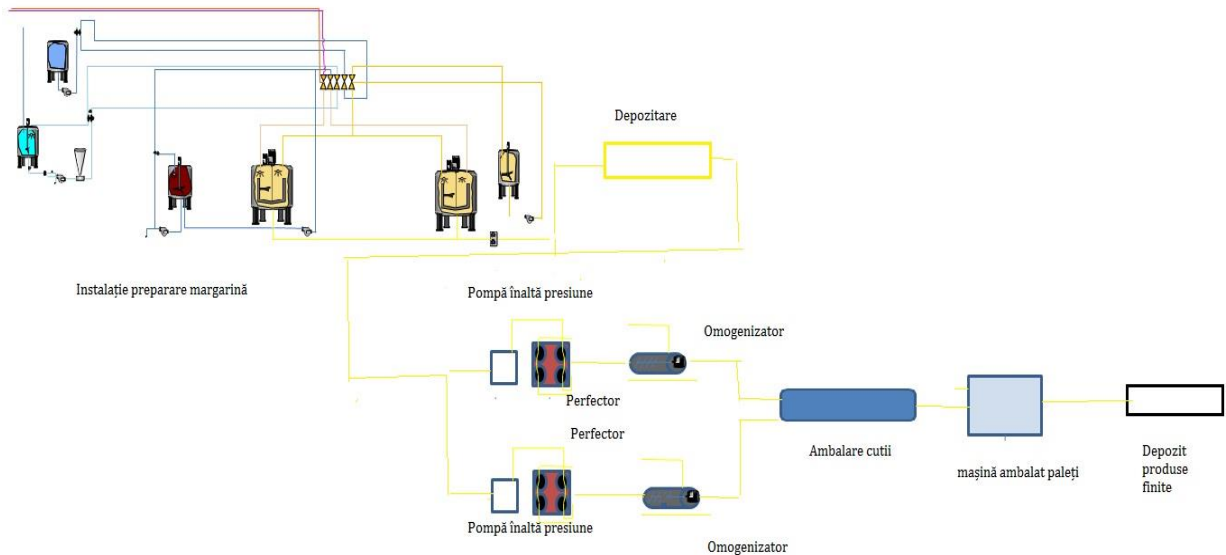


Figura 4.14. Schema liniei flexibile de fabricație a margarinei – Unilever România

Descrierea sistemului HibriPlanJSS

Sistemul HibriPlanJSS este dezvoltat în limbajul de programare C++, utilizând platforma QtCreator, versiunea 2.6.2 și a fost testat pe o unitate de calcul cu următoarea configurație: AMD FX™-6100 Six-Core Processor, 3.30 GHz CPU, având la baza metoda hibridă propusă HibriPSO-SA.

Interfața sistemului prezentată în **Figura 4.15**, este intuitivă și ușor de utilizat chiar și de către un utilizator nespecializat.



Figura 4.15 Interfața sistemului HibriPlanJSS

TESTAREA ALGORITMULUI HIBRIPSO-SA

În continuare, metoda hibridă care stă la baza sistemului de planificare HibriPlanJSS a fost testată în condițiile unei linii de fabricație reale din cadrul Unilever România specializată pentru producția de margarină.

Structura liniei flexibile de fabricație care constituie obiectul studiului de caz este prezentată în **Figura 4.14**. Este de asemenea cunoscut faptul că pe această linie de fabricație pot fi produse 8 sortimente de margarină, fiecare dintre acestea realizându-se după o rețetă diferită și fiind ambalate în cutii de greutate și formă diferită.

În **Tabelul 4.4** sunt prezentate 10 seturi de date reprezentând cerințe diferite de producție pentru cele 8 sortimente de margarină ce pot fi produse pe linia de fabricație considerată.

În **Tabelul 4.5** sunt prezentate cele 20 de seturi de date de test pentru parametrii specifici metodei hibride HibriPSO-SA luați în considerare: T_0 – temperatura inițială, α – rata de răcire, Np - numărul de particule disponibile, $\phi_1 \in (0,1)$ - parametrul cognitiv, $\phi_2 \in (0,1)$ - parametrul social, I_{max} - numărul maxim de iterații.

Fiecare din cele 10 seturi de date pentru cerințe de producție a fost testat împreună cu fiecare din cele 20 de seturi de date de test pentru parametrii specifici metodei hibride (**Tabelul 4.5**).

Tabelul 4.4 Cerințe diferite de producție

Setul de date	Numărul de tone/produs							
	S1	S2	S3	S4	S5	S6	S7	S8
DS2_1	10	12	12	10	12	10	10	10
DS2_2	30	35	50	30	40	35	30	30
DS2_3	40	0	0	40	40	0	0	0
DS2_4	0	40	40	0	0	40	40	40
DS2_5	40	40	40	40	0	0	0	0
DS2_6	0	0	0	0	30	30	30	30
DS2_7	30	20	30	20	30	20	30	20
DS2_8	10	20	30	40	30	20	10	20
DS2_9	40	10	20	50	20	30	30	40
DS2_10	50	50	50	50	50	50	50	50

4.5 ANALIZA COMPARATIVĂ A METODEI PROPUSE PENTRU UNILEVER

În **Tabelul 4.6** sunt prezentate rezultatele obținute de cei 5 algoritmi considerați pentru valorile funcției f .

Tabelul 4.6 Rezultate comparative referitoare la valoarea funcției f

Setul de date	Funcția fitness f					
	Hibri PSO-SA	SA	PSO	ACO	AG	FCFS
DS2_1	0.027356	0.027813	0.028288	0.029501	0.028823	0.029958
DS2_2	0.103311	0.132082	0.120080	0.145796	0.012907	0.129679
DS2_3	0.041554	0.055455	0.045964	0.069179	0.058219	0.055964
DS2_4	0.089335	0.119225	0.090120	0.010295	0.090442	0.092676
DS2_5	0.064086	0.064086	0.064086	0.065412	0.065106	0.064086
DS2_6	0.045349	0.055585	0.046942	0.049081	0.047369	0.047926
DS2_7	0.077623	0.078394	0.079177	0.080132	0.080257	0.087338
DS2_8	0.080702	0.082502	0.081625	0.082967	0.082144	0.089643
DS2_9	0.092063	0.092447	0.096471	0.096471	0.141612	0.124124
DS2_10	0.246820	0.308497	0.256843	0.259705	0.256843	0.308497

Funcția obiectiv f este cea care măsoară calitatea soluției obținute și conform ecuației (1.5) din capitolul 1 această funcție trebuie minimizată. Astfel conform rezultatelor experimentale din **Tabelul 4.6** se poate observa faptul că cele mai mici valori pentru funcția f sunt obținute în cazul aplicării metodei hibride propuse în 100% din cazuri.

Rezultatele experimentale prezentate în **Tabelul 4.7** evidențiază faptul că metoda hibridă propusă reușește să obțină valori mult mai bune comparativ cu ceilalți algoritmi luați în considerare în 90% din cazuri (în 10% este obținută aceeași valoare pentru C_{max} ca în cazul algoritmului SA), iar comparativ cu datele preluate de la Unilever în 100% din cazuri.

În **Tabelul 4.8** sunt prezentate rezultatele obținute în urma rulării celor 6 algoritmi referitoare la timpul total necesar rulării. Se evidențiază astfel faptul că deși timpul necesar rulării asociat metodei hibride nu este la fel de mic ca în cazul algoritmilor de o complexitate mai mică (cum ar fi SA și FCFS), comparativ cu celelalte metode valoarea obținută nu este foarte ridicată.

Tabelul 4.7 Rezultate comparative referitoare la valoarea C_{max}

Setul de date	C_{max} (minute)						
	Hibri PSO-SA	SA	PSO	ACO	AG	FCFS	Unilever
DS2_1	1710	1740	1770	1840	1820	1870	1840
DS2_2	5390	5860	5430	5950	5440	5820	5530
DS2_3	2610	3090	2730	3440	3230	3120	2770
DS2_4	4520	5600	4570	4640	4580	4720	4610
DS2_5	3510	3510	3560	3580	3560	3510	3690
DS2_6	2670	3140	2720	2790	2780	2820	2770
DS2_7	4010	4040	4080	4130	4130	4470	4610
DS2_8	4140	4220	4180	4210	4200	4570	4150
DS2_9	5470	5590	5600	5600	6010	5670	5540
DS2_10	8150	8870	8280	8290	8280	8870	9230

Un alt criteriu luat în considerare pentru evaluarea calității soluțiilor obținute metoda hibridă pusă în discuție se referă la productivitatea medie (măsurată în numărul de tone produse pe oră), iar rezultatele experimentale sunt prezentate în **Tabelul 4.9**.

Se poate observa faptul că planul de producție obținut prin intermediul metodei hibride duce la o creștere sesizabilă a productivității medii comparativ cu datele reale obținute de la Unilever, dar și cu ceilalți algoritmi luați în considerare.

Tabelul 4.9 Rezultate comparative referitoare la productivitatea medie

Setul de date	Productivitate medie (tone/ora)						
	Hibri PSO-SA	SA	PSO	ACO	AG	FCFS	Unilever
DS2_1	2.807	2.759	2.712	2.609	2.637	2.567	2.600
DS2_2	2.672	2.457	2.652	2.420	2.647	2.474	2.600
DS2_3	2.759	2.330	2.637	2.093	2.229	2.308	2.600
DS2_4	2.655	2.143	2.626	2.586	2.620	2.542	2.600
DS2_5	2.652	1.368	1.348	1.341	1.348	1.368	2.600
DS2_6	2.697	2.293	2.647	2.581	2.590	2.553	2.600
DS2_7	2.993	2.970	2.941	2.906	2.906	2.685	2.600
DS2_8	2.609	2.559	2.584	2.565	2.571	2.363	2.600
DS2_9	2.633	2.576	2.571	2.571	2.396	2.540	2.600
DS2_10	2.945	2.706	2.899	2.895	2.899	2.706	2.600

O reprezentare grafică a rezultatelor obținute în ceea ce privește productivitatea medie pentru cei 6 algoritmi propuși cât și pentru datele reale obținute de la Unilever este prezentată în **Figura 4.21**.

În acest capitol algoritmul HibriPSO-SO a fost testat în condiții diferite, atât pentru o serie de date de test teoretice reprezentate de 15 benchmark-uri recunoscute în literatura de specialitate, cât și în contextul unei linii de fabricație reale disponibile la Unilever România.

Datele experimentale au fost supuse unor comparații pe baza mai multor criterii de evaluare a performanței algoritmului propus, iar rezultatele susțin faptul că metoda hibridă propusă se poate aplica cu succes în domeniul planificării automate în linii flexibile de fabricație.

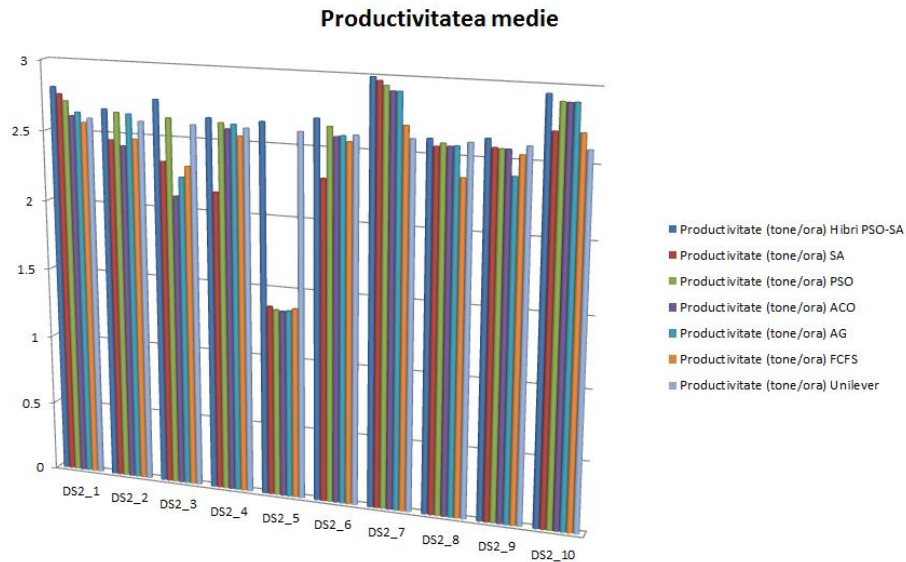


Figura 4.21 Evoluția valorilor productivității medii

CONCLUZII FINALE ȘI DIRECȚII VIITOARE DE CERCETARE

Complexitatea problemei planificării producției, în special în ceea ce privește domeniul liniilor flexibile de fabricație, a constituit o adevărată provocare pentru cercetătorii din întreaga lume, lucru care a dus la încercarea de a rezolva această problemă printr-o multitudine de metode și tehnici care propun soluționarea ei într-o manieră optimală.

Direcțiile de cercetare actuale sunt orientate către adoptarea unor tehnici hibride, bazate pe algoritmi specifici inteligenței artificiale, care încearcă să îmbine avantajele strategiilor de căutare locală cu cele ale căutării globale pentru determinarea unor soluții optime și eficiente pentru problema propusă.

În *Capitolul 1* a fost studiat domeniul planificării automate a producției, atât din punctul de vedere al formulării problemei de planificare, cât și prin evidențierea particularităților acestora pentru liniile flexibile de fabricație. De asemenea acest capitol include o prezentare a viziunii sistemice asupra planificării automate a producției, precum și detalierea modelului matematic al problemei JSS ce descrie un sistem flexibil de fabricație.

În ultima parte a acestui capitol este inclus un studiu asupra sistemelor de planificare disponibile atât la nivel național, cât și la nivel internațional, pentru care au fost luate în considerare o serie de sisteme de planificare dezvoltate atât pentru mediul academic, cât și pentru mediul comercial, cu aplicabilitate în domenii diverse ale industriei.

Se remarcă faptul că majoritatea sistemelor studiate sunt potrivite pentru a fi aplicate în cadrul sistemelor flexibile de fabricație, pentru problema JSS. Un sistem de planificare de calitate trebuie să pună la dispoziția utilizatorului o interfață prietenoasă, să fie adaptabil pentru sisteme de fabricație diferite, să fie rapid, să optimizeze utilizarea mașinilor disponibile, să evite supraaglomerarea depozitelor ș.a.m.d. Toate aceste cerințe, adăugate la faptul că problema JSS este una NP-complexă, duc la concluzia că studiul în acest domeniu trebuie continuat, în speranța obținerii unor metode mai bune de planificare, pentru a optimiza din ce în ce mai mult întregul proces de producție.

Capitolul 2 conține o prezentare succintă a tehnicilor de Inteligență Artificială selectate (Particle Swarm Optimization, Ant colony Optimization, Algoritmi genetici și Simulated Annealing). Pseudocodul acestora este prezentat atât în forma clasică, cât și în varianta adaptată pentru problema JSS pe baza modelului matematic inclus în capitolul 1. De asemenea pentru fiecare tehnică în parte a fost propus un exemplu de problemă rezolvată pentru a evidenția mai bine modul de reprezentare a soluției, stările inițiale și intermediare, precum și modalitatea de calcul a funcției fitness aferente.

Sinteza lucrărilor științifice publicate pe această temă în literatura de specialitate din ultimii ani oferă o imagine de ansamblu asupra stadiului actual al cercetărilor în acest domeniu și evidențiază interesul tot mai mare pentru identificarea unei soluții optime pentru JSS, în special prin utilizarea unor tehnici hibride bazate pe Inteligență Artificială.

În *Capitolul 3* a fost studiat comportamentul sistemelor de planificare bazate pe tehnici specifice Inteligenței Artificiale Computaționale dezvoltate pentru problema JSS atât în contextul unor date de test fictive, cât și în contextul unor date de test clasice menționate în literatura de specialitate. În urma testelor efectuate s-a evidențiat faptul că sistemul de planificare bazat pe PSO a obținut cele mai bune rezultate în 73.33% din cazuri pentru datele de test fictive și în 86.66 % din cazuri pentru datele de test clasice. Sistemul bazat pe ACO a obținut cele mai bune rezultate în 26.66% din cazuri pentru

datele de test fictive și în 20% din cazuri pentru datele de test clasice, iar sistemul bazat pe AG a obținut cele mai bune rezultate în 53.33% din cazuri pentru datele de test fictive și în 0% din cazuri pentru datele de test clasice.

În aceste condiții algoritmul PSO a fost selectat pentru a fi parte componentă a metodei hibride deoarece a obținut rezultatele cele mai bune într-un număr de cazuri de test semnificativ mai mare decât celelalte două, rămânând ca variantă alternativă algoritmul ACO ce s-a clasat pe locul doi din punctul de vedere al calității rezultatelor obținute. AG a obținut rezultate bune pentru datele de test fictive, însă rezultate foarte slabe pentru datele de test clasice, iar rezultatele obținute pentru aceste date de test sunt esențiale. Metoda hibridă propusă încearcă să îmbine avantajele tehnicii selectate (PSO) cu cele ale SA și să le adapteze pentru rezolvarea problemei JSS.

Metoda HibriPSOSA aduce o serie de elemente de noutate cum ar fi: modalitatea prin care este construită populația inițială, aplicarea SA pentru un set de indivizi selectați aleator la fiecare pas al algoritmului PSO, implementarea unei funcții pentru identificarea eventualelor discrepante ce apar în cadrul planificării pentru a asigura fezabilitatea soluțiilor, precum și utilizarea unei funcții fitness complexe pentru o evaluare cât mai corectă a soluțiilor candidat. Pseudocodul metodei hibride este detaliat în acest capitol și descrie atât modalitatea de funcționare, cât și datele de intrare și de ieșire specifice.

Validarea metodei HibriPSO-SA s-a realizat pe baza unui set de 15 benchmark-uri specifice JSS recunoscute în literatura de specialitate, iar rezultatele experimentale subliniază faptul că în cazul soluțiilor obținute timpul total necesar execuției reușește să se încadreze în intervalul determinat de limita inferioară și limita superioară acceptate în 100% din cazuri. De asemenea în 73.33% din cazurile considerate, algoritmul HibriPSO-SA reușește să obțină un plan de fabricație al cărui timp total necesar execuției este egal cu valoarea optimă recunoscută în literatura de specialitate.

În *Capitolul 4* sunt prezentate cercetările experimentale privind dezvoltarea sistemului de planificare a producției în linii flexibile de fabricație HibriPlanJSS bazat pe tehnica hibridă HibriPSO-SA. Mai întâi a fost prezentată metodologia utilizată pentru dezvoltarea acestui sistem de planificare, apoi a fost detaliată structura acestuia.

Studiul de caz a fost realizat la Unilever România pe o linie flexibilă de fabricație corespunzătoare producției de margarină. Formalismul rețelilor Petri este utilizat pentru a modela acțiunile ce presupun utilizarea în comun a unor resurse pentru linia flexibilă de fabricație disponibilă la Unilever, oferind o vedere de ansamblu asupra modului de funcționare a acesteia. Rezultatul studiului de caz a fost implementarea sistemului de planificare HibriPlanJSS bazat pe metoda hibridă propusă ce pune la dispoziția utilizatorilor o interfață prietenoasă, ușor de folosit, ce oferă facilitățile necesare pentru planificarea producției în funcție de comenzile primite. Acest sistem este dezvoltat în limbajul de programare C++, utilizând platforma QtCreator, versiunea 2.6.2 și a fost testat pe o unitate de calcul cu următoarea configurație: AMD FX™-6100 Six-Core Processor, 3.30 GHz CPU.

Etapa de testare a metodei a avut la baza datelor preluate de la Unilever România unde este disponibilă o linie flexibilă de fabricație specializată pe producția de margarină. Au fost luate în considerare zece seturi de date de test reprezentând cerințe de producție diferite pentru cele opt sortimente de margarină ce pot fi produse acolo.

Rezultatele au fost comparate și analizate pe baza unor criterii legate de calitatea soluției (măsurată prin valoarea funcției obiectiv f), timpul total necesar planificării (C_{max}), timpul necesar rulării algoritmului, precum și productivitatea medie obținută.

Conform rezultatelor experimentale obținute pentru datele de test reale preluate de la Unilever România, metoda hibridă propusă a obținut cele mai bune valori privind calitatea soluției, timpul total necesar planificării, precum și pentru productivitatea medie pentru 90% din toate cazurile luate în considerare. În 10% din cazuri au fost obținute valori egale cu cele obținute prin metoda SA. În ceea ce privește timpul necesar rulării algoritmului, metodele mai puțin complexe (SA și FCFS) au obținut din nou cele mai bune valori. Însă comparativ cu metodele mai complexe (AG, PSO, ACO), timpul necesar rulării metodei hibride păstrează aceleași valori rezonabile.

În urma analizei comparative a rezultatelor experimentale, se poate concluziona faptul că metoda hibridă propusă se poate aplica cu succes în domeniul planificării automate în linii flexibile de fabricație. Din datele analizate se remarcă faptul că metoda propusă obține cele mai bune valori pentru toate criteriile de comparație luate în

considerare, atât în cazul benchmark-urilor clasice considerate, cât și în cazul setului de date real preluat de la Unilever România.

Contribuțiile originale ale acestei teze de doctorat sunt următoarele:

- Abordarea sistemică a problemei de planificare în linii flexibile de fabricație în secțiunea 1.3, Figura 1.3;
- Prezentarea modelului matematic detaliat specific problemei JSS în secțiunea 1.5;
- Identificarea obiectivelor principale ale problemei de planificare și integrarea acestora în formula de calcul a funcției fitness f prezentată în ecuația 1.5;
- Realizarea unui studiu bibliografic detaliat privind stadiului actual al cercetărilor în domeniul planificării automate a producției, precum și o analiză comparativă a soluțiilor existente atât pe plan național cât și internațional (secțiunea 1.7);
- Adaptarea algoritmilor PSO, ACO, AG și SA pentru JSS pe baza modelului matematic specific în secțiunile 2.1-2.4;
- Identificarea tendințelor actuale în literatura de specialitate în domeniul soluționării eficiente a problemei JSS utilizând metode specifice IA sau tehnici hibride (secțiunea 2.2, secțiunea 2.3);
- Testarea algoritmilor PSO, AG și ACO atât pe baza unui set de 15 benchmark-uri clasice, cât și pe baza unor date de test fictive, urmată de analiza comparativă a rezultatelor în vederea selectării celui mai potrivit algoritm pentru a fi parte integrantă a metodei hibride (secțiunea 3.1.3).
Criteriile de comparație au fost următoarele:
 - în cazul benchmark-urilor: încadrarea în intervalul acceptat, atingerea valorii optime și timpul de rulare (secțiunea 3.1.1);
 - în cazul datelor de test fictive: valoarea funcției f , valoarea C_{max} , timpul de rulare și numărul de soluții (secțiunea 3.1.2);
- Propunerea metodei hibride bazate pe PSO și SA în secțiunea 3.2;

- Formularea ecuației funcției obiectiv (ecuația 1.5 din capitolul 1, ecuația 3.1 din capitolul 3)
- Validarea algoritmului hibrid HibriPSO-SA cu ajutorul a 15 benchmark-uri clasice din literatura de specialitate (secțiunea 3.3);
- Detalierea metodologiei utilizate pentru dezvoltarea sistemului de planificare (secțiunea 4.1);
- Abordarea sistemică a structurii sistemului inteligent de planificare HibriPSO-SA în secțiunea 4.2;
- Modelarea sistemului de fabricație specializat pentru producția de margarină disponibil la Unilever România utilizând Rețele Petri (secțiunea 4.3);
- Implementarea sistemului de planificare HibriPlanJSS pentru studiul de caz realizat la Unilever (secțiunea 4.4);
- Testarea metodei hibride pentru datele de producție reale preluate de la Unilever România (secțiunea 4.2.);
- Analiza comparativă a rezultatelor obținute atât de metoda hibridă, cât și de algoritmi specifici Inteligenței Artificiale Computaționale (PSO, ACO și AG) în secțiunea 4.3.

Principalele direcții viitoare de cercetare vor avea ca subiect extinderea modelului JSS prin includerea defecțiunilor mașinilor apărute în timpul procesului de fabricație, precum și testarea metodei hibride propuse în contextul unor linii flexibile de fabricații multiple, ce utilizează în comun anumite resurse. De asemenea, se vor aplica tehnici de data mining în cadrul algoritmului hibrid, pentru selectarea particulelor ce vor fi modificate utilizând tehnica Simulated Annealing, în vederea îmbunătățirii performanțelor acestuia.

BIBLIOGRAFIE

1. Adams, J., Balas, E., Zawack, D., *The shifting bottleneck procedure for job shop scheduling*, Management Science, vol. 34, nr. 3, p. 391-401, 1988.
2. Affenzeller, M., Winkler, S., Wagner, S., Beham, A., *Genetic Algorithms and Genetic Programming. Modern Concepts and Practical Applications*, ISBN 978-1-58488-629-7, CRC Press, Taylor & Francis Group, 2009
3. Akers S.B., Friedman J., *A non numerical approach to scheduling problems*, Operations Research, vol. 3, 1955, p. 429 – 442.
4. Alba, E., Dorronsoro, B., *Cellular Genetic Algorithms*, ISBN978-0-387-77609-5, Springer, 2008
5. Applegate, D., Cook, W., *A computational study of the job-shop scheduling instance*, ORSA Journal on Computing, vol. 3, p. 149-156, 1991.
6. Arezoo, B., Ridgway, K., Al-Ahmari, A.M.A., *Selection of cutting tools and conditions of machining operations using an expert system*, Computers in Industry, Vol. 42, Issue 1, pp 43-58, June 2000, ISSN 0166-3615
7. Artiba, A., Elmaghraby, S.E., *The Planning and Scheduling of Production Systems. Methodologies and applications*, ISBN 978-1-4612-8507-6, Chapman&Hall, 1997
8. Asadzadeh L., *A local search genetic algorithm for the job shop scheduling problem with intelligent agents*, Computers & Industrial Engineering, Vol. 85, pp. 376-383, July 2015, ISSN 0360-8352,
9. Assaf, R., *Job Shop Lean Production Implementation Using Program Evaluation and Review Techniques (PERT)*, Proceedings of the 2014 International Conference on Industrial Engineering and Operations Management, Bali, Indonesia, p. 1554-1560, 7-8 Ianuarie 2014.
10. Baptiste, L., Flamini, M., Sourd, F., *Lagrangian bounds for just-in-time job-shop*, Computers & Operations Research, Vol. 35, Issue 3, p. 906-915, 2008
11. Barenji A. V., Barenji R. V., Hashemipour M., *Flexible testing platform for employment of RFID-enabled multi-agent system on flexible assembly line*, Advances in Engineering Software, Vol. 91, , pp. 1-11, January 2016, ISSN 0965-9978,
12. Bartak, R., Salido, M.A., Rossi, F., *Constraints Satisfaction Techniques in Planning and Scheduling*, Journal of Intelligent Manufacturing, Vol. 21, Issue 1, p. 5-15, 2010.
13. Bellman, R.E., *An introduction to Artificial Intelligence: Can Computers Think?*, Boyd&Fraser Publishing Company, 1978
14. Blazewicz, J., Ecker, K.H., Pesch, E., Schmidt. G., Weglaz, J., *Handbook on Scheduling. From Theory to Applications*, ISBN: 978-3-540-28046-0, Springer-Verlag Berlin Heidelberg, 2007
15. Blum, C., *Ant Colony Optimization: Introduction and recent trends*, Physics of Life Review, vol. 2, p. 353-373, 2005
16. Bonabeau, E., Dorigo, M., Theraulaz, T., *From Natural to artificial Swarm Intelligence*, Oxford University Press, New York, 1999
17. Bondrea I., *Proiectarea asistată tehnologică*, Sibiu, 2009

18. Bonet, P., Llado, C.M., Puijaner, R., Knottenbelt, W.J., *PIPE v2.5: A Petri Net Tool for Performance Modelling*, Proceedings of the 23rd Latin American Conference on Informatics (CLEI 2007), San Jose, Costa Rica, October 2007.
19. Booth, A.M., *Job Shop Scheduling using Multi-Agent Systems*, Adaptive Computing in Design and Manufacture VI, ISBN 978-1-85233-829-9, p. 149-160, 2004
20. Boukef, H., Benrejeb, M., Borne, P., *Flexible Job-shop Scheduling Problems Resolution Inspired from Particle Swarm Optimization*, Studies in Informatics and Control, ISSN 1220-1766, vol. 17 (3), p. 241-252, 2008.
21. Bowman, E. H., *The schedule-sequencing problem*, Operations Research, vol. 7, pp. 621 – 624, 1959.
22. Cerny, V., *A Thermodynamical Approach to the Travelling Salesman Problem: An Efficient Simulation Algorithm*, Journal of Optimization Theory and Applications, Vol. 45, p. 41-51, 1985.
23. Chakraborty, U.K., *Computational Intelligence in Flow Shop and Job Shop Scheduling*, ISBN 978-3-642-02835-9, Springer-Verlag Berlin Heidelberg, 2009.
24. Chakraborty, S., Bhowmik, S., *An Efficient Approach to Job Shop Scheduling Problem using Simulated Annealing*, International Journal of Hybrid Information Technology, Vol.8, No.11 pp.273-284, 2015.
25. Chen, W.C., Tseng, S.S., Wang, C.Y., *A novel manufacturing defect detection method using association rule mining techniques*, Expert Systems with Applications 29: pp. 807–815, 2005
26. Chong, C.S., Low, M.Y.H, Sivakumar, A.I., Gay, K.L., *A Bee Colony Optimization Algorithm to Job Shop Scheduling*, Proceedings of the Winter Simulation Conference, Monterey, ISBN 1-4244-0500-92006, p. 1954-1961, 2006.
27. Chrystolouris, G., *Manufacturing Systems. Theory and Practuce*, ISBN 978-1-4757-2215-4, Springer, 1992
28. Chtourou, H., Wassim M., and Aref M., *An expert system for manufacturing systems machine selection*, Expert Systems with Applications, 28.3 pp. 461-467, 2005
29. Clerc., M., *Particle Swarm Optimization*, ISTE Ltd, ISBN: 978-1-905209-04-0 London, UK, 2006.
30. Corzalezany, M.B., *Computation Intelligence Systems and Applications. Neuro-Fuzzy and Fuzzy Neural Synergisms*, ISBN: 3-7908-1439-3, Shysica-Verlag Heidelberg Germany, 2002.
31. Cruz-Chávez, M.A., Martínez-Rangel, M.G., Cruz-Rosales, M.H., *Accelerated simulated annealing algorithm applied to the flexible job shop scheduling problem*, International Transactions in Operational Research, John Wiley & Sons Ltd, p. 1-19, 2015.
32. Cupek R., Ziebinski A., Huczala L., Erdogan H., *Agent-based manufacturing execution systems for short-series production scheduling*, Computers in Industry, Vol. 82, pp. 245-258, October 2016,
33. Dingle, N.J., Knottenbelt, W.J. Suto, T., *PIPE2: A Tool for the Performance Evaluation of Generalised Stochastic Petri Nets*, ACM SIGMETRICS Performance Evaluation Review (Special Issue on Tools for Computer Performance Modelling and Reliability Analysis), Vol. 36(4), pp. 34-39, March 2009,

34. Dorigo, M., Blum, C., *Ant Theory Optimization theory : A survey*, *Theoretical Computer Science*, Vol. 344, nr. 3-4, p. 243-278, Elsevier ,2005.
35. Dorigo, M., Brittari, M., *Ant Colony Optimization and swarm intelligence*, Springer, 2004
36. Dorigo, M., Stutzle, T., *Ant Colony Optimization: Overview and Recent Advantages*, IRIDIA-Technical Report Series, Tehnical Repor No. TR/IRIDIA/2009-013, Mai 2009.
37. Dorigo, M., Stutzle, T., *The Ant Colony Optimization metaheuristics: Algorithms, applications and advances*, International Series in Operations Research & Management Sciences, Handbook of metaheuristic, Vol. 57, p. 250-285, Springer, 2003.
38. Drăgoi G., Guran M. – *Sisteme integrate de producție asistate de calculator*, Editura Tehnică, București, 1997.
39. Dreo, J., Petrowski, A., Siarry, P., Taillard, E., *Metaheuristics for Hard Optimization. Simulated Annealing, Tabu Search, Evolutionary and Genetic Algorithms, Ant Colonies. Methods and Case Studies*, ISBN-13 3-540-23022-9, Springer Berlin Heidelberg New York, 2006.
40. Duch, W., Mandziuk, J., *Challenges for Computational Intelligence*, ISBN: 978-3-540-71983-0, Springer-Verlag Berlin Heidelberg, 2007.
41. Emmons, H., Vairaktarakis, G., *Flow Shop Scheduling. Theoretical Results, Algorithms and Applications*, ISBN: 978-1-4614-5151-8, Springer Science+Business Media New York, 2013.
42. Engelbrecht, A. P., *Fundamentals of Computational Swarm Intelligence*, John Wiley & Sons, ISBN:0470091916, Chichester, UK, 2005.
43. Eswaramurthy, V.P., *Tabu Search Strategies for Solving Job Shop Scheduling Problems*, Journal of Advanced Manufacturing Systems, Vol. 06, No. 59, 2007
44. Eswaramurthy, V.P., Tamilarasi, A., *Hybridizing tabu search with ant colony optimization for solving job shop scheduling problems*, International Journal of Advanced Manufacturing Technology, Vol.40, Issue 9-10, p. 1004-1015, 2010.
45. Farughi, H., Yegane, B.Y., Fathian, M., *A new critical path method and memetic algorithm for flexible job shop scheduling with overlapping operations*, Simulation: Transactions of Society for Modelling and Simulation International, Vol. 89, No. 2, p. 264-277, 2012.
46. Fisher, H., Thompson, G.L., *Probabilistic learning combinations of local job-shop scheduling rules*, J.F. Muth, G.L. Thompson (eds.), Industrial Scheduling, Prentice Hall, Englewood Cliffs, New Jersey, pp. 225-251, 1963.
47. Florez, E., Gomez, W., Bautista, L., *Ant Colony Optimization Algorithm for Job Shop Scheduling Problem*, International Journal of Artificial Intelligence & Application, Vol. 4, No. 4, p. 56-66, 2013.
48. Gao, K.Z., Suganthan, P.N., Chua, T.J., Chong, C.S., Cai, T.X., Pan, Q. K., *A two-stage artificial bee colony algorithm scheduling flexible job-shop scheduling problem with new job insertion*, Vol. 42, Issue 21, p. 7652-7663, 2015.
49. Gao, K.Z., Suganthan, P.N., Pan, Q.K., Chua, T.J., Chong, C.S., Cai, T.X., *An improved artificial bee colony algorithm for flexible job-shop scheduling problem with fuzzy processing time*, Expert Systems and Applications, Vol. 65, p. 52-67, 2016.

50. Gao, L., Pan, Q.K., *A suffeled multi-swarm micro-migrating birds optimizer for a multi-resource-constrained flexible job shop scheduling problem*, Information Sciences, Vol. 372, p. 655-676, 2016.
51. Ge, H., Du, W, Qian, F, *A Hybrid Algorithm Based on Particle Swarm Optimization and Simulated Annealing for Job shop Scheduling*, Proceedings of the Third International Conference on Natural Computation, Vol. 3, p. 715-719, 2007.
52. Gen, M., Gheng, R., *Genetic Algorithms and engeneering optimization*, Wiley Series in Engineering Design and Automation, Series Editor, 2000.
53. Ghallab, M., Nau, D., Traverso, P, *Automated Planning. Theory and practice*, Elsevier, 2014.
54. Glover, F., *Future Paths for Integer Programming and Links to Artificial Intelligence*, Computer and Operations Research, Vol. 13, p. 533-549, 1986.
55. Goldberg, D.E., *The Design of Innovation. Lessons from and for Competent Genetic Algorithms*, ISBN 978-1-4757-3645-8, Springer, 2002.
56. Gonzalez, M. A., Vela, C.R., Varela, R., *A New Hybrid Genetic Algorithm for the Job Shop Scheduling Problem with Setup Times*, Proceedings of the Eighteenth International Conference on Automated Planning and Scheduling (ICAPS 2008), pp116-123. 2008.
57. Gopi, E.S., *Algorithm Collection for Digital Signal Processing Applications using Matlab*, ISBN 978-1-40206409-8, Springer, 2007
58. Groover, M. P. , *Automation Systems and Computer Integrated Manufacturing*, Second Edition, 2003.
59. Guo, Z.X., Wong, W.K., Leung, S.Y.S., Fan, J.T., *Intelligent production control decision support system for flexible assembly lines*, Expert Systems with Applications, Vol. 36, Issue 3, Part 1, pp. 4268-4277, April 2009, ISSN 0957-4174,
60. Heizer, J., *Production and Operations Management*, Allyn and Macon, Needham Heights, Massachusetts, 1991.
61. Herrmann, J.W., *Handbook of Production Scheduling*, ISBN: 0-387-33115-8, Springer Science+Business Media Inc., 2006
62. Hong-Wei, G., Wen-Li, D., Feng, Q., Lu, W., *An Intelligent Hybrid Algorithm for Job-Shop Scheduling Based on Particle Swarm Optimization and Artificial Immune System*, Advances in Soft Computing , Volume 41, pp. 628-637, 2007.
63. Iazewicz, J.B., Pesch, E., Igorzata Sterna, M., *The disjunctive graph machine representation of the job shop scheduling problem*, European Journal of Operational Research, Vol. 127, Issue 2, p. 317-331, 2000.
64. Ispas, C., Mohoră, C., Caramihai, S., Călin, O., *Simularea sistemelor integrate de fabricație*, ISBN 973-9493-15-7 , Editura Bren, București, 1999.
65. Jamili, A., Shafia, M. A., Tavakkoli-Moghaddam, R., *A Hybrid Algorithm based on Particle Swarm Optimization and Simulated Annealing for Periodic Job Shop Scheduling Problem*, The International Journal of Advanced Manufacturing Technology, ISSN 0268-3768, vol. 54 (1-4), pp. 309-322, 2010.
66. Jarbouli, B., Siarry,P., Teghem, J., *Metaheuristics for Production Scheduling*, Automation-Control and Industial Engineering Series, ISBN: 978-84821-497-2, ISTE ltd, SUA, 2013

67. Karaboga, D., Basturk, B., *A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm*, Journal of Global Optimization, vol. 39, nr. 3, pp. 459-471, 2007
68. Kennedy, J., Russel, E., *Swarm Intelligence*, Academic Press, 2001
69. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P., *Optimization by Simulated Annealing*, New Series, Vol. 220, No. 4598, p. 671-680, 1983
70. Kong, X., Liu, S., Wang, Z., *A new hybrid artificial bee colony algorithm for global optimization*, International Journal of Computer Science Issues, Vol. 10, Issue 1, Nr. 1, ISBN: 1694-0784, 2013
71. Korytkowski, P., Rymaszewski, S., Wiśniewski, T. *Ant Colony Optimization for job shop scheduling using multi-attribute dispatching rules*, The International Journal of Advanced Manufacturing Technology, Vol. 67, Issue 1, p. 231-241, 2013.
72. Kouider, A., Bouzouia, B., *Multi-agent job shop scheduling system based on cooperative approach of idle time minimization*, International Journal of Production research, Vol. 50, Issue 2, p. 409-424, 2012.
73. Kuncheva, L, Porter, T., *Computational Intelligence: Methods and Applications*, International ICSC Congress, Bangor, Wales, United Kingdom, 19-22 Iunie, 2001.
74. Lawrence, S., *Resource constrained project scheduling: an experimental investigation of heuristic scheduling techniques (Supplement)*, Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh, Pennsylvania, 1984
75. Lee W.-C., Wang J.-Y., *A three-agent scheduling problem for minimizing the makespan on a single machine*, Computers & Industrial Engineering, Available online 6 February 2017, ISSN 0360-8352,
76. Leon, F., *Metode de planificare – notițe de curs*, Universitatea Tehnică „Gh. Asachi” Iasi, 2010
77. Li, J., Pan, Q., Xie, S., Wang, S., *A hybrid artificial bee colony algorithm for flexible job shop scheduling problem*, International Journal of Computers, Communications & Control, Vol. IV, No. 2, p. 286-296, 2011.
78. Li, K., Leung, J.Y.-T., Cheng, B.-Y., *An agent-based intelligent algorithm for uniform machine scheduling to minimize total completion time*, Applied Soft Computing, Vol. 25, , pp. 277-284, December 2014, ISSN 1568-4946,
79. Liu, L.L., Hu, R.S., Hu, X.P., Zhao, G.P., Wang, S., *A hybrid PSO-GA algorithm for job shop scheduling in machine tool production*, International Journal of Production Research, Volume 53, Issue 19, pp. 5751-5781, 2015
80. Liu, Z., *Investigation of Particle Swarm Optimization for Job Shop Scheduling Problem*, Third International Conference on Natural Computation ICNC 2007, Vol. 3, p. 799-803
81. Lopez P., Roubellat F., *Production Scheduling*, ISTE Ltd, 2008
82. Maghfiroh, M. F. N., Darmawan, A., Yu, V.F., *Genetic Algorithm for Job Shop Scheduling Problem: A Case Study*, International Journal of Inoovation, Management and Technology, Vol. 4, No. 1, p. 137-140, 2013.
83. Manezzo, V., Gamberdella, L., Luigi, F., *Ant Colony Optimization*, 2004
84. Maroosi, A., Muniyanfi, R.C., Sundararajan, E., Zin, A.M., *A parallel membrane inspired harmony search for optimization problems: A case study based on flexible job shop scheduling problem*, Applied Soft Computing, Vol. 49, p. 120-136, 2016.

85. Mattfeld, D.C., *Evolutionary Search and the Job Shop. Investigation on Genetic Algorithms for Production Scheduling*, ISBN 978-7908-0917-6, Springer, 1996.
86. McCarthy, J., *What is Artificial Intelligence?*, Standford University, 2007.
87. Mesghouni, K., Hammadi, S., Bone, P., *Evolutionary algorithms for Job Shop Scheduling*, International Journal of Applied Mathematics and Computer Science, vol. 14, nr.1, p. 91-103, 2004.
88. Metaxiotis, K. S., Askounis, D., Psarras, J., *Expert systems in production planning and scheduling: A state-of-the-art survey*, Journal of Intelligent Manufacturing, Vol. 13, Issue 4, p. 253-260, 2002.
89. Michalewicz, Z, Fogel, D.B., *How to solve it: Modern Heuristics*, Springer, 2004.
90. Mohora, C. Coteș, C.E., Pătrașcu, G., *Simularea sistemelor de producție*, ISBN 973-2708-68-9, Editura Agir, București, 2001.
91. Monette, J.N., Deville, Y., Van Hentenryck, P., *Just-In-Time Scheduling with Constrains Programming*, The 19th International Conference on Automated Planning and Scheduling ICAPS, Thesaloniki, p. 241-248, 19-23 Septembrie 2009.
92. Nahmias, S., *Production and Operation Analysis*, Santa Clara University, ISBN 0072865385, 2005
93. Nahmias, S., *Production and Operations Analysis*, 6th ed, Mc Graw Hill, 2009.
94. Nakandhrakumar,R.S, Balachandar, M., *Implementation of Simulated Annealing Technique for Optimizing Job Shop Scheduling Problem*, International Journal of Advanced Mechanical Engineering. ISSN 2250-3234, Vol. 4, No. 2, p. 169-174, 2014.
95. Nazif, H., *Solving Job Shop Scheduling Problem Using an Ant Colony Algorithm*, Journal of Asian Scientific Research, Vol. 5, No. 5, p. 261-268, 2015.
96. Neagu, C., Rosu, M.M., Catana, M., *A New Approach to Scheduling and Control of Job-Shop Production*, Scientific Bulletin U.P.B Bucuresti, Seria D, Vol. 68, No. 3, p. 37-44, 2006
97. Neagu, G., *A Multi-agent Model for Job Shop Scheduling*, Studies in Computational Intelligence, Service Orientation in Holonic and Multi-Agent Manufacturing Control, Vol. 402, p. 211-225, ISBN 978-3-642-27448-0, 2012.
98. Nicoara, E.S., Filip, F.G., Paraschiv, N., *Simulation-based Optimization using Genetic Algorithms for multi-objective Flexible JSSP*, Studies in Informatics and Control, ISSN 1220-1766, vol. 20 (4), p. 333-344, 2011.
99. Nouri, M., Bekrar, A., Jemai, A., Niar, S., Ammari, A. C., *An effective and distributed particle swarm optimization algorithm for flexible job-shop scheduling problem*, Industrial Engineering and Systems Management, Proceedings of 2013 International Conference on Publication Rabat, p. 1-6, 28-30 Octombrie 2013.
100. Oprea, M., Nicoara, S., *Inteligenta artificiala*, Universitatea Petrol-Gaze din Ploiesti, 2005
101. Oprea, M., *Sisteme bazate pe cunoștințe*, Editura Matrix Rom, Bucuresti, 2002
102. Othman S. B., Zgaya H., Dotoli M., Hammadi S., *An agent-based Decision Support System for resources' scheduling in Emergency Supply Chains*, Control Engineering Practice, Vol. 59, pp. 27-43, February 2017, ISSN 0967-0661,
103. Paraschiv, N., *Sisteme în timp real – notițe de curs*, Ploiesti, 2010.

104. Păstrăvanu, O., Matcovschi, M., Mahulea, C., *Aplicații ale Rețelelor Petri în studierea sistemelor cu evenimente discrete*, Ed. Gh. Asachi, 2003.
105. Peng, B., Lu, Z., Cheng, T.C.E., *A Tabu Search/Path Relinking Algorithm to Solve the Job Shop Scheduling Problem*, Computer & Operations Research, Vol. 53, p. 154-164, 2015.
106. Peter B. , *Scheduling Algorithms*, ISBN 978-3-540-69515-8, Springer, 2007.
107. Pinedo, M.L., *Planning and Scheduling in Manufacturing and Services*, Springer Series in Operatio Research, ISBN 9780387273990, 2005.
108. Poli, R., Kennedy, J., Blackwell, T., *Particle swarm optimization. An overview*. Swarm Intelligence, Vol. 1, nr. 1, pp. 33-57, 2007.
109. Pongchairerks, P. , Kachitvichyanukul, V., *A Particle Swarm Optimization Algorithm on Job Shop Scheduling Problem with Multi-Purpose Machine*, Asia-Pacific Journal of Operational Research, Vol. 2, Issue 2, p.161-184, 2009.
110. Pongchairerks, P., *Particle Swarm Optimization algorithm applied to scheduling problems*, Science Asia Journal, Vol. 35, No. 1, p. 89-94, 2009.
111. Ponnambalam, S. G., Jawahar, N., Girish, B. S., *An Ant Colony Optimization Algorithm for Flexible Job Shop Scheduling Problem*,New Advanced Technologies, Aleksandar Lazinica (Ed.), InTech, ISBN 978-953-307-067-4, p. 73-92, 2010.
112. Ponnambalam, S.G., Aravindan, P., Rajesh, S.V., *A TabuSearch Algorithm for Job Shop Scheduling*, The International Journal on Advanced Manufacturing Technology, Vol. 16, Issue 10, p. 765-771, 2000
113. Poole, D.L., Mackworth, A.K., *Artificial Intelligence. Foundations of Computational Algorithms*, ISBN: 978-0-521-51900-7, Cambridge University Oress, 2010
114. Qiu, X., Lau, H., *An AIS-based hybrid algorithm for static job shop scheduling problem*, Journal of Intelligent Manufacturing, Volume 25, Issue 3, pp. 489-503, 2014.
115. Raeesi, N.M., Kobti, Z., *A memetic algorithm for job shop scheduling using a critical-path-based local search heuristic*, Memetic Computing, Vol. 4, Issue 3, p. 231-245, 2012.
116. Rego, C., Alidaee, B., *Metaheuristic Optimization via Memory and Evolution Tabu Search and Scatter Search*, Kluwer Academic Publisher, ISBN 1-4020-8134-0, 2005
117. Ren, Q., Wang, Y., *A new hybrid genetic algorithm for job shop scheduling problem*, Computer & Operation Research, Volume 39, Issue 10, pp. 2291-2299, 2012
118. Rey, G.Z., Bekrar, A., Trentesaux, D., Zhou, B.H., *Solving the flexible job-shop just-in-time scheduling problem with quadratic earliness and tardiness costs*,The International Journal of Advanced Manufacturing Technology, Vol. 81, Issue 9, p. 1871-1891, 2015
119. Rulm, W., Do, M.,B., Fromherz, M., *On-line Planning and Scheduling for High-Speed Manufacturing*, Proceedings of the Fifteenth International Conference on Automated Planning and Scheduling ICAPS, Monterey, Californiam, USA, 5-10 Iunie 2005.
120. Ruml, W., Do, M.B., *On-line Planning and Schedulin: An Application to Controlling Modular Printers*,Journal of Artificial Intelligence Research, vol. 40, p. 415-468, 2011.
121. Russel, S.J., Norving, P., *Artificial Intelligence. A modern approach*”, Pentice Hall, Englewood Cliffs, New Jersey, 1995.

122. Sawik T., *Production Planning and scheduling in flexible assembly systems*, Springer, 1999
123. Sha, D.Y., Hsu, C.Y., *A hybrid particle swarm optimization approach for the job shop scheduling problem*, Computer and Industrial Engineering, Vol.51, Issue 4, p. 791-808, 2006.
124. Sha, D.Y., Lin, H.H., *A multi-objective PSO for job shop scheduling problems*, Expert Systems with Applications, Vol. 37, Issue 2, p. 1065-1070, 2010.
125. Shen W., Hao Q., Yoon H. J., Norrie D., *Applications of agent-based systems in intelligent manufacturing: An updated review*, Advanced Engineering Informatics, Vol. 20, Issue 4, pp. 415-431, October 2006, ISSN 1474-0346,
126. Siddique, A., Nazmul, H., *Computational Intelligence: Synergies of Fuzzy Logic, Neural Networks and Evolutionary Computing*, John Weil & Sons, 2013.
127. Song, C.L., Liu, X.B., Wang, W., Xin, B., *A Hybrid Particle Swarm Optimization Algorithm for Job Shop Scheduling Problem*, International Journal of Advancements in Computing Technology, vol. 3 (4), p. 79-88, 2011.
128. Song, X., Cao, Y., Chang, C., *A hybrid Algorithm of PSO and SA for Solving JSP*, Proceedings of the Fifth International Conference on Fuzzy Systems and Knowledge Discovery, ISBN 978-0-7695-3305-6, vol. 1, p. 111-115, 2008.
129. Song, X., *Hybrid particle swarm algorithm for job shop scheduling problems*, Future Manufacturing Systems , ISBN 978-953-307-128-2, Chapter 12, pp. 235-268, 2010.
130. Sotskov, Y., Gholami, O., *Mixed graph model and algorithms for parallel- machine job-shop scheduling problems*, International Journal of Production Research, In press, p. 1-16, 2015
131. Soutrakul, N,m Nachtmann, H.L., Cassady, C.R., *Genetic Algorithms for integrated preventive maintenance planning and production scheduling for single machine*, Computers in Industry, Vol. 56, nr. 2, p. 161-168, 2005.
132. Srinivas, M.K.T. , Allada, V. , *Solving the machine loading problem in a flexible manufacturing system using a combinatorial auction-based approach*, International Journal of Production Research, Vol. 42, No. 9, p. 1879-1893, 2004.
133. Stevenson, W., *Introduction to Management Science*, 2nd ed., Richard D. Irwin, Burr Ridge, Illinois, 1992.
134. Storer, R.H. , Wu, S.D., Vaccari, R. *New search spaces for sequencing instances with application to job shop scheduling*, Management Science, Vol. 38, p. 1495-1509, 1992.
135. Stutzle, T., Dorigo, M., *A short convergence proof for a class of ACO algorithms*, Transactions on Evolutionary Computation, Vol. 6, nr. 4, pp. 358-365,IEEE, 2002.
136. Sule, D.R., *Production Planning and Industrial Scheduling. Examples, Case Studies and Applications*, Second Edition, ISBN 1-4200-4420-6, CRC Press TaYLOR & Francis Group LLC, SUA, 2008.
137. Sumathi, S., Hamsapriya, T., Surekha, P., *Evolutionary Intelligence. An Intoroduction to Theory and Applications with Matlab*, ISBN: 978-3-540-75158-8, Springer-Verlag Berlin Heidelberg, 2008.

138. Symeonidis, A.L., Kehagias, D.D., Mitkas, P.A., Intelligent policy recommendations on enterprise resource planning by the use of agent technology and data mining techniques, *Expert Systems with Applications* 25, pp. 589–602, 2003
139. Taillard, E. *Benchmarks for basic scheduling problems*. *European Journal of Operational Research*, Vol. 64, p. 278-285, 1993.
140. Tamas, P., Iles, B., Tollar, S., *Simulation of Flexible Manufacturing Systems*, *Advanced Logistic Systems*, Vol. 6, No. 1, p. 25-32, 2012.
141. Tamilarasi, A., Anantha Kumar, T., *An enhanced genetic algorithm with simulated annealing for job shop scheduling*, *International Journal of Engineering, Science and Technology*, Vol. 2, No. 1, p. 144-151, 2010.
142. Thang, J., Zhang, G., Lin, B., Zhang, B., *A Hybrid Algorithm for Job Shop Scheduling Problem*, *Procedia Engineering*, Volume 15, pp. 3678-3683, 2011.
143. Tibrewal, R., Chang, P. C., Tiwari, M. K., *Production planning optimization for manufacturing and remanufacturing system in stochastic environment*, *Journal of Intelligent Manufacturing*, vol. 24, nr. 4, pp 717-72, 2013.
144. Tripathi, A., *Multi Agent system in Job Shop Scheduling using Contract Net Protocol*, *International Journal of Computer Applications*, Vol. 94, No. 16, p. 24-29, 2014.
145. Umar, A.U., Ariffin, K.A., Ismail, N., Tang, S.H., *Hybrid multiobjective genetic algorithms for integrated dynamic scheduling and routing of jobs and automated-guided vehicle (AGV) in flexible manufacturing systems (FMS) environment*, *International Journal of Advanced Manufacturing Technology*, vol. 91, issue 9, p 2123-2141, 2015.
146. Van Laarhoven, P.J., Aarts, E.H., *Simulated Annealing: Theory and Application*, Kluwer Academic Publishing, 1992.
147. Wagner H. M., *An integer programming model for machine scheduling*, *Naval Research Logistics Quarterly*, Vol. 6, p. 131 – 140, 1959.
148. Wang S., Wang L., A knowledge-based multi-agent evolutionary algorithm for semiconductor final testing scheduling problem, *Knowledge-Based Systems*, Vol. 84, , pp. 1-9, August 2015, ISSN 0950-7051,
149. Wang, H., Li, Y., Wang, J., Bi, G., Zhang, Z., A design Strategy for Multi-agent based Industrial Intelligent Monitor Processing, *Applied Mechanics and Materials*, vol 224, pg. 184 – 188, 2012.
150. Wang, L., Shen, W., *Process Planning and Scheduling for Distributed Manufacturing*, ISBN: 978-1-84628-751-0, Springer-Verlag London Limited, 2007.
151. Wang, L., Zhou, G., Xu, Y., Wang, S., Liu, M., *An effective artificial bee colony algorithm for flexible job-shop scheduling problem*, *The International Journal of Advanced Manufacturing Technology*, Vol. 60, Issue 1, p. 303-315, 2012.
152. Watson, J.P., Beck, J.C., Howe, A., Whitley, L.D., *Problem difficulty for tabu search in job-shop scheduling*, *Artificial Intelligence*, vol. 143, Issue 2, Elsevier, p. 189-217, 2003
153. Werner, F. *Genetic Algorithms for Shop Scheduling Problems: A Survey*, Preprint, Vol. 11, p. 1-66, 2011.
154. Winston, P.H., *Artificial Intelligence*, Addison Wesley, 3rd Edition, 1992.

155. Wooldridge, M., *An introduction to multiagent systems (2nd ed.)*. Wiley Publishing, 2009.
156. Xhafa, F., Abraham, A., *Metaheuristics for Scheduling in Industrial and Manufacturing Applications*, ISBN: 978-3-540-78984-0, Springer-Verlag Berlin Heidelberg, 2008.
157. Xia, W. J., Wu, Z. M., *An effective hybrid optimization approach for multi-objective flexible job shop scheduling problems*, *Computers & Industrial Engineering*. Volume 48, pp. 409-425, 2005.
158. Xing, L.N., Chen, Wang, P., Zhao, Q.S., Xiong, J., *A Knowledge Based Ant Colony Optimization for Flexible Job Shop Scheduling Problems*, *Applied Soft Computing*, Vol. 10, Issue 3, p. 888-896, 2010.
159. Yamada, T., Nakano R., *A genetic algorithm applicable to large-scale job-shop instances*, R. Manner, B. Manderick (eds.), *Parallel instance solving from nature 2*, North-Holland, Amsterdam, p. 281-290, 1992.
160. Yang, X.S., Cui, Z., Xiao, R., Gandomi, A.H., Karamanoglu, M., *Swarm Intelligence and Bio-Inspired Computation*, ISBN: 978-0-12-405163-8, Elsevier, 2013.
161. Yang, X.S., *Nature-Inspired Metaheuristic Algorithms*. Second Edition., ISBN: 978-1-905986-28-6, Luniver Press, 2010.
162. Yazdani, M. , Gholami, M., Zandieh, M., Mousakhani, M. , *A Simulated Annealing Algorithm for Flexible Job-Shop Scheduling Problem*, *Journal of Applied Sciences*, Vol. 9, p. 662-670, 2009.
163. Yin, J., Chen, B., *The simulation expert system for job shop on-line scheduling based on G2*, *Proceedings of the 16th International Conference on Industrial Engineering and Engineering Management IE&EM 09*, Beijing, ISBN 978-1-4244-3671-2, p. 860-863, 2009
164. Zhang, Q., Manier, H., Manier, M.A., *A Modified Disjunctive Graph for Job Shop Scheduling Problems with Bounded Processing Time and Transportations Constraints*, *14th IFAC Symposium Information Control Problems in Manufacturing*, Vol. 45, Issue 6, p. 1377-1282, 23-25 Mai 2012.
165. Zhang, R., Song, S. Wu, C., *A two-stage hybrid particle swarm optimization algorithm for the stochastic job shop scheduling problem*, *Knowledge Based Systems*, Vol. 27, p. 393-206, 2012.
166. Zhang, R., Wu, C., *An Artificial Bee Colony Algorithm for the Job Shop Scheduling Problem with Random Processing Time*, *Entropy*, Vol. 12, No. 12, 2011.
167. Zhu, Z.C., Ng, K.M., Ong, H.L., *A modified tabu search algorithm for cost-based job shop problem*, *Journal of the Operational Research Society*, Vol. 61, No. 4, p. 611-619, 2010.
168. Zobolas, C.I., Trantilis, C.D., Ioannou, G., *A hybrid evolutionary algorithm for the job shop scheduling problem*, *Journal of the Operational Research Society*, Vol. 60, No. 2, p. 221-235, 2009.

WEBOGRAFIE

169. ***, *Sistemul RVHolon*, http://www.rvholon.cimr.pub.ro//index_ro.html, accesat pe data de 15 ianuarie 2011
170. ***, *AIMMS*, <http://www.aimms.com/>, accesat pe 25 noiembrie 2010
171. ***, *Asprova*, <http://www.asprova.com/>, accesat pe 10 februarie 2011
172. ***, *Cros ERP*,
http://www.iooss.ro/software/servicii_software/solutii_erp, accesat pe 5 aprilie 2010
173. ***, *Lekin Software*, <http://www.stern.nyu.edu/om/software/lekin/index.htm>, accesat pe data de 25 noiembrie 2010
174. ***, *LFAC ECLIPS*,
http://www.ac.tuiasi.ro/grant_CNFIS0006/raport1an.htm, accesat pe 15 aprilie 2010
175. ***, *LFAC FMC Driver*, http://www.ac.tuiasi.ro/grant_CNFIS0006/raport_final.htm accesat pe 15 aprilie 2010
176. ***, *SIM System*,
<http://www.simtechcorp.com/flexiblemanufacturing.html>, accesat pe data de 20 decembrie 2010
177. ***, *Sistemul RVHolon*, http://www.rvholon.cimr.pub.ro//index_ro.html, accesat pe data de 15 ianuarie 2011
178. ***, *Sitemul SofHicor*, http://www.sofhicor.cimr.pub.ro/index_ro.htm, accesat pe data de 15 ianuarie 2011
179. ***, *Tupas Software* , <http://www.tuppas.com/advanced-planning-software/advanced-planning-software.htm>, accesat pe data de 25 ianuarie 2011
180. ***, *CSP2SAT: JSS benchmark results*, <http://bach.istc.kobe-u.ac.jp/csp2sat/jss/>, accesat la data de 18.01.2015.
181. ***, *Fișa produsului program Cross ERP*, http://bnp-web.ici.ro:8090/reports/rwservlet?fis_pdf&PRODUS=131, accesat la data de 03.11.2016.
182. ***, *Solutia ASP*, <http://www.seniorsoftware.ro/aps-planificare-productie/>, accesat la data de 01.11.2016.
183. ***, *Senior Software*, <http://www.seniorsoftware.ro/despre-companie-senior-software/>, accesat la data de 01.11.2016.
184. ***, *Soluția CHARISMA ERP – Modulul Producție*,
<http://www.charisma.ro/solutii-si-sisteme-erp/productie/> , accesat la data de 02.11.2016.
185. ***, *Sistemul SAS FAIR VALUE ERP*,
<http://fairvalue.ro/produse/erp/managementul-productiei/>, accesat la data de 20.10.2016.

186. ***, *Sistemul ASCENT MRP*, <http://www.ascenterp.ro/industrii/erp-productie/>, accesat la data de 20.10.2016.
187. ***, *Sistemul CLARVISION ERP*, <http://clarvision.ro/solutii/clarvision-erp/>, accesat la data de 20.10.2016.
188. ***, *Sistemul SAP FAIR VALUE ERP*, <http://jobpack.com/>, accesat la data de 21.10.2016.
189. ***, *Sistemul IQMS Manufacturing ERP*,
<http://www.iqms.com/products/erp/manufacturing/scheduling/>, accesat la data de 21.10.2016.
190. ***, *Sistemul PLEX*, <http://www.plex.com/products/manufacturing-operations-management-mom/advanced-planning-production-scheduling-software.html>, accesat la data de 21.10.2016.
191. ***, *Sistemul SeikiSoftware Advanced Manufacturing*,
<http://www.seikisystems.co.uk/product/production-planning-and-scheduling/>, accesat la data de 22.10.2016.
192. ***, *Sistemul Preactor*,
http://www.preactor.com/Home.aspx#.WBzBu_mLS70, accesat la data de 22.10.2016.
193. ***, *Sistemul PIMSS*, <https://www.mjc2.com/production-planning-software.htm>, accesat la data de 23.10.2016.
194. ***, *Sistemul Demand Solutions*,
<http://www.demandsolutions.com/advanced-planning-production-planning-scheduling-software.html>, accesat la data de 23.10.2016.
195. ***, *Sistemul Tricorn Systems*, <http://www.tricornsystems.co.uk/>, accesat la data de 23.10.2016.
196. ***, *Sistemul Giraffe Production Systems*,
<http://www.automationmag.com/directory/lean-manufacturing/flow/jit/106-giraffe-production-systems-pty-ltd/view-details.html>, accesat la data de 24.10.2016.
197. ***, *Sistemul SEMS*, <http://www.esteelman.com/index.php/en/sems-system/production-advanced-scheduling-system>, accesat la data de 24.10.2016.
198. ***, *Artificial Bee Colony Algorithm (ABC)Homepage*, <http://mf.erciyes.edu.tr/abc/>, accesat la data de 3.11.2016