

# ORGANIZAȚIA VIRTUALĂ GRIDMOSI - COMPONENTĂ A INFRASTRUCTURII NAȚIONALE DE CERCETARE

Gabriel Neagu  
[gneagu@ici.ro](mailto:gneagu@ici.ro)

Neculai Andrei  
[nandrei@ici.ro](mailto:nandrei@ici.ro)

Vasile Sima  
[vsima@ici.ro](mailto:vsima@ici.ro)

Alexandru Stanciu  
[stanciu@grid.ici.ro](mailto:stanciu@grid.ici.ro)

*Institutul Național de Cercetare - Dezvoltare în Informatică, ICI, București*

Nicolae Țăpuș  
[ntapus@cs.pub.ro](mailto:ntapus@cs.pub.ro)

Valentin Cristea  
[valentin@cs.pub.ro](mailto:valentin@cs.pub.ro)

*Universitatea Politehnica București*

Cătălin Nae  
[cnae@aero.incas.ro](mailto:cnae@aero.incas.ro)

*Institutul Național de Cercetări Aerospațiale București*

Rodica Potolea  
[rodica.potolea@cs.utcluj.ro](mailto:rodica.potolea@cs.utcluj.ro)

*Universitatea Tehnică din Cluj-Napoca*

Dana Petcu  
[petcu@info.uvt.ro](mailto:petcu@info.uvt.ro)

*Universitatea de Vest din Timișoara*

**Rezumat:** Infrastructura Grid reprezintă o componentă esențială a unei infrastructuri moderne de cercetare. Principalele clase de resurse ale acestei infrastructuri sunt, pe de o parte site-urile Grid conectate prin linii de mare viteză și care oferă capacități de calcul și de memorare, iar pe de altă parte, aplicațiile de înaltă performanță din diverse domenii (științifice, tehnice, economice), a căror execuție eficientă este condiționată de existența unor resurse de calcul semnificative. Proiectul GridMOSI are ca obiectiv constituirea acestor două clase de resurse pentru domeniul modelării, simulării și optimizării de înaltă performanță. Accesul comunității potențialilor utilizatori la aceste resurse este asigurat conform conceptului de organizație virtuală în tehnologie Grid. Lucrarea prezintă oferta de soluții MOSI a proiectului, care valorifică expertiza de excelență existentă la nivelul instituțiilor partenere. Domeniile abordate sunt: modelare și conducere avansată cu calculator, optimizare fără restricții, modelare numerică complexă și optimizare MDO în aplicații industriale, optimizare bazată pe algoritmi evolutivi, planificarea taskurilor în Grid folosind algoritmi genetici, algoritmi criptografici și criptanalitici.

**Cuvinte cheie:** infrastructură electronică, Grid, organizație virtuală, modelare, simulare, optimizare.

## 1. Introducere

Modelarea, simularea și optimizarea reprezintă abordări folosite pe scară largă de comunitatea științifică în proiectarea și operarea sistemelor complexe construite de către om, în investigarea sau îmbunătățirea proceselor tehnice, economice ori biologice. Creșterea complexității sistemelor de analizat sau proiectat necesită platforme de calcul, algoritmi și programe asociate foarte fiabile și eficiente. Elaborarea unor astfel de instrumente implică costuri extrem de mari, cât și expertiză în mai multe domenii. Abordarea Grid este una dintre cele mai promițătoare direcții de acțiune pentru rezolvarea unor probleme de calcul de mari dimensiuni într-un mod aparent necostisitor. Conceptul Grid se referă în principal la coordonarea și distribuția transparentă a capacității de rezolvare a problemelor la nivelul organizațiilor virtuale, dinamice, cu caracter multi-instituțional. Principalul obiectiv al tehnologiilor Grid este de a permite conexiunea și colaborarea pe scară largă a mediilor de calcul distribuite, indiferent de gradul lor de complexitate sau eterogenitate.

Avantajele pe care aceste tehnologii le oferă comunității academice, de cercetare și de afaceri se referă la: accesul la serviciile de prelucrare distribuită a datelor pe resurse TI partajate, orientarea resurselor disponibile în fiecare organizație către rezolvarea problemelor și obiectivelor specifice și mai puțin către upgradarea continuă a infrastructurii informatice, diminuarea dificultăților generate în fiecare organizație de lipsa de specialiști capabili să administreze infrastructuri din ce în ce mai sofisticate, furnizarea unor premise favorabile pentru comunicarea și colaborarea globală.

La nivel european, Grid și tehnologiile asociate reprezintă un pilon esențial al suportului de tip *infrastructură electronică* (eInfrastructure) necesar implementării conceptului de *Arie Europeană de Cercetare* [1]. Conform punctului de vedere al *eInfrastructure Reflection Group* – organism de elaborare a politicilor de dezvoltare a eInfrastructure la nivel european, infrastructura electronică include trei niveluri arhitecturale: rețea, resurse (de calcul, de memorare, de culegere date), middleware (servicii specializate de acces la resurse, securitate, contabilizare, statistici). Împreună cu nivelul resurselor de tip aplicații constituie

*infrastructura Grid*. Pentru țara noastră, infrastructura Grid este considerată o componentă esențială a societății bazate pe cunoaștere [2]. În sfârșit, conceptul de *organizație virtuală* (VO) se referă la partajarea flexibilă, sigură, coordonată a resurselor între colecții dinamice de indivizi, instituții și resurse [3]. Profilul VO este dat de orientarea aplicațiilor Grid înscrise în organizația respectivă.

Proiectul GridMOSI reprezintă o soluție de valorificare a potențialului tehnologiilor Grid în beneficiul comunității academice și de cercetare prin oferirea a două clase de resurse specifice infrastructurii de cercetare: *soluții de modelare, simulare și optimizare* (MOSI) în diverse domenii aplicative și *centre de resurse Grid* instalate și administrate la nivelul instituțiilor partenere.

Lucrarea prezintă oferta de soluții MOSI abordată în cadrul proiectului și structurată pe domeniile aplicative vizate de activitatea de cercetare din instituțiile partenere: *modelare și conducere avansată cu calculator și optimizare fără restricții* (ICI), *modelare numerică complexă și optimizare MDO în aplicații industriale* (INCAS), *optimizare bazată pe algoritmi evolutivi* (UVT), *planificarea taskurilor în Grid folosind algoritmi genetici* (UPB-CNTI), *optimizarea unor algoritmi criptografici și criptanalitici* (UTCN). Selectarea acestor domenii și a soluțiilor / aplicațiilor asociate s-a făcut pe baza următoarelor criterii: expertiză de excelență pentru domeniul respectiv la nivelul colectivului autor al propunerii, stadiul de elaborare al soluției propuse în tehnologie clasică; argumentarea opțiunii de migrare în mediul Grid din punct de vedere al câștigului de performanță și accesibilitate a soluției.

## 2. Modelare și conducere avansată cu calculator

În afara domeniilor tradiționale, cum ar fi ramurile industriale de vârf (aeronautică și astronautică, electronică, industria de automobile, industria chimică și farmaceutică, etc.), modelarea sistemică a pătruns și în domeniile netehnice, cum ar fi biologia sau finanțele. Datorită importanței lor au fost propuse o mare varietate de metode numerice pentru rezolvarea acestor probleme. Există, de asemenea, numeroase implementări în programe de calcul asociate (de exemplu în MATLAB, SLICOT, Scilab), dar acestea se adresează rezolvării unor probleme de mici dimensiuni, pe mașini seriale. Un exemplu de mediu de rezolvare a problemelor de la distanță, folosind Internet, este sistemul NEOS, elaborat relativ recent și întreținut de Argonne National Laboratory, Argonne, Illinois, S.U.A.

O abordare practică în modelarea proceselor, *identificarea sistemelor*, constă în determinarea modelelor utilizând înregistrările existente ale variabilelor cheie ale procesului. Modelele obținute sunt adesea prea mari pentru a fi folosite efectiv pentru conducerea proceselor sau pentru proiectarea unor regulatoare adecvate. Mai mult, regulatoarele proiectate sunt uneori de ordin prea mare pentru a fi folosite în timp real. De aceea, este foarte important să se aplice tehnici de reducere a ordinului modelelor sau reguletoarelor. În fine, sunt necesare tehnici, proceduri și metodologii de proiectare pentru obținerea legilor de conducere satisfăcând cerințe practice specificate, incluzând stabilitatea și performanța în buclă închisă. Adesea, aceste tehnici se bazează pe optimizare neliniară.

În cadrul proiectului sunt abordate probleme privind identificarea parametrilor și simularea unor sisteme dinamice liniare și a anumitor clase de sisteme neliniare, reducerea ordinului modelelor, optimizare neliniară, care se înscriu în direcțiile de interes major pe plan mondial. În domeniul identificării sistemelor este valorificată experiența participării la elaborarea, evaluarea și întreținerea bibliotecii de programe SLICOT-Subroutine Library in Control Theory (V. Sima), care se bazează pe bibliotecile BLAS (Basic Linear Algebra Subprograms) și LAPACK (Linear Algebra Package), iar parțial pe extensiile lor la mașini cu memorie distribuită, PBLAS (Parallel BLAS) și ScaLAPACK (Scalable LAPACK).

Aplicația Grid de **reducere a ordinului modelelor** include algoritmi operând pe modele de sisteme dinamice liniare invariante în timp, descrise în spațiul stărilor. Se garantează că eroarea (măsurată în norma infinit) dintre matricea de transfer a modelului original și a modelului de ordin redus este mai mică decât o toleranță prestabilită. Sunt incluse metode de balansare și trunchiere, aproximații prin perturbații singulare și aproximații în norma Hankel. Există mai multe implementări în versiunea clasică ale procedurilor de reducere a ordinului modelelor, atât în mediul de lucru MATLAB, cât și în biblioteca SLICOT, care include și unele implementări pe calculatoare paralele distribuite, utilizând nucleele paralele din ScaLAPACK și PBLAS și bibliotecile seriale LAPACK și BLAS. Aceste implementări vor fi transpuse și experimentate în mediul Grid. Problematika de bază a calculului de performanță în automată este tratată în [4, 5] și în lucrările citate acolo. Nu există actualmente variante algoritmice adecvate pentru implementări pe calculatoare paralele pentru toate subproblemele implicate în rezolvarea problemelor de reducere a ordinului.

Aplicația Grid de **determinare a modelelor matematice** include algoritmi operând pe datele de intrare-ieșire existente sau generate, prin experimente dedicate, ale variabilelor sistemelor sau proceselor dinamice. Într-o primă etapă, aplicația va genera modele de sisteme multivariabile liniare discrete, invariante în timp,

descrie în spațiul stărilor. Ordinul sistemului poate fi impus de utilizator, poate fi determinat automat de aplicație, sau poate fi selectat de utilizator în funcție de rezultatele furnizate de aplicație pentru o mulțime specificată de valori. Sunt incluse opțiuni pentru evaluarea calității modelelor determinate, în vederea ghidării utilizatorului pentru alegerea celei mai potrivite valori a ordinului sistemului. Din punct de vedere tehnic, se vor folosi abordări specifice de determinare a unui subspațiu de stare, în speță, MOESP (*Multivariable Output Error state SPace*), N4SID (*Numerical algorithm for Subspace State Space System Identification*) și combinația acestora.

Există mai multe implementări în versiunea clasică ale procedurilor de identificare a sistemelor, atât în mediul de lucru MATLAB (în special, în System Identification Toolbox), cât și în biblioteca SLICOT. Implementarea din SLICOT, realizată de V. Sima, va fi transpusă și experimentată în mediul Grid. Această implementare, operațională pe calculatoare seriale, oferă mai multe opțiuni pentru selectarea metodei folosite (MOESP, N4SID sau combinația acestora) sau a rezultatelor care trebuie obținute (toate sau o parte a matricilor sistemului, matricile de covarianță, matricea predictorului Kalman, estimăția stării inițiale prin două tehnici diferite). Alte aspecte, teoretice sau practice, sunt prezentate în [5, 6-8] și în lucrările citate acolo. Nu se cunosc implementări în mediul Grid ale algoritmilor și procedurilor de identificare, dar problematica modelării este tratată în sistemele NetSolve și GridSolve.

Migrarea în mediul Grid este justificată de complexitatea mare de calcul a problemelor de identificare a sistemelor, generată de: numărul de variabile de intrare și ieșire ale unor procese (fizice, chimice, biologice, financiare etc.) și de numărul de eșantioane privind evoluția variabilelor de intrare și ieșire.

Pentru cazul sistemelor neliniare, soluția MOSI propusă în proiect este dedicată sistemelor de tip Wiener, care constau dintr-o parte liniară și o neliniaritate statică. Pentru determinarea părții neliniare se vor folosi algoritmi Levenberg-Marquardt specializați, care exploatează structura particulară a problemei. Și în acest caz va fi utilizată ca referință implementarea din SLICOT. Această implementare oferă opțiuni pentru selectarea metodei folosite (algoritmi specializați de gradienti conjugați, sau utilizând factorizări Cholesky sau QR exploatănd structura) sau a modului de inițializare a algoritmilor (toate variabilele, sau numai cele corespunzătoare părții liniare sau neliniare) [9].

### 3. Optimizare fără restricții

Domeniul optimizării, cunoscut încă sub forma de programare matematică pentru a-l separa de cel al optimizării dinamice și al controlului optimal, implică cunoașterea unui aparat matematic foarte evoluat care se reflectă în codurile care implementează algoritmi de programare matematică. În lucrarea [10] se prezintă în detaliu principalele clase de probleme de optimizare și software-ul asociat lor. Datorită caracteristicilor problemelor de optimizare este foarte dificil și extrem de costisitor de a instala, utiliza și compara, la nivel profesional, diferite programe de optimizare. Aceasta justifică efortul elaborării unui mediu interactiv, în tehnologie Grid, în care utilizatorul poate apela diferite coduri de optimizare.

Dintre formele de reprezentare a modelelor de programare matematică, programarea liniară este cea mai utilizată în aplicațiile practice. Aceasta se datorează în primul rând aspectelor teoretice care rezultă din caracterul liniar al modelului, ceea ce implică scufundarea întregii probleme în clasa problemelor de algebră liniară computațională, una dintre cele mai studiate ramuri ale analizei numerice. În al doilea rând există posibilitatea ca cel puțin în primă aproximație, clase foarte largi de fenomene din diverse domenii de activitate ca tehnică, economie, business, administrație, biologie etc. să poată fi reprezentate ca modele liniare. În același timp, dezvoltările foarte spectaculoase în ceea ce privește modelarea fenomenelor ca probleme de programare liniară a făcut posibilă translatarea acestui corp de cunoștințe în domeniul optimizării neliniare unde s-au obținut rezultate deosebit de valoroase, atât în ceea ce privește modelarea prin intermediul limbajelor de programare matematică, cât și rezolvarea acestor modele (Andrei [10-12]).

În rezolvarea unei probleme de programare liniară se observă că există o dihotomie clară între algoritmist și modelist (utilizator sau client), cu ținte și dorințe foarte diferite. Interesul major manifestat de algoritmist constă în rezolvarea unor chestiuni legate de analiza modelului din punctul de vedere al soluției (admisibile, inefabile sau nemărginite), simplificarea modelului în sensul eliminării restricțiilor redundante sau a unor variabile calculate direct din marginile simple, elaborarea de algoritmi și implementarea lor în programe eficiente, studiul sensibilității soluției la variația parametrilor modelului. Modelistul este interesat în conceptualizarea și formularea, adică reprezentarea în termeni matematici, a porțiunii de univers în care este interesat. Având în vedere structura foarte simplă a modelului de programare liniară rezultă că și comunicarea acestuia serverului de optimizare este mai simplă. Se cunosc două proceduri de asamblare a unei probleme de programare liniară:

- a. MPS (Mathematical Programming Systems): clientul generează un fișier cu problema în formatul MPS și îl transmite serverului, unde se găsesc instalate o serie de pachete profesionale de programare liniară (ca de ex. ASLO [13], MINOS [14]), care admit la intrare fișierul cu problema în formatul MPS;
- b. limbaje de programare matematică: cele mai utilizate sunt GAMS [15] și AMPL [16], precum și limbajul de optimizare liniară ALLO [17]. Serverul conține translatorul asociat fiecărui limbaj în parte pentru formatul MPS.

Descrierea externă permite elaborarea de prototipuri în sensul că modelul matematic este descris în format algebric, complet independent de date. Modificările de structură ale modelului se execută în prototip, cele referitoare la date - în baza de date asociată prototipului.

În cazul optimizării neliniare, un mod de abordare evoluat constă în utilizarea unui limbaj de programare matematică în care se exprimă problema și care, prin intermediul unui translator, generează o descriere internă a modelului direct acceptată de un optimizator profesional. Cele mai importante limbaje de programare matematică care au capabilități de rezolvare a problemelor de optimizare neliniară sunt GAMS și AMPL.

**Biblioteca de programe de înaltă performanță pentru optimizare fără restricții** conține o multitudine de pachete de programe și subrutine specializate dedicate rezolvării modelelor de optimizare fără restricții. Se are în vedere includerea de algoritmi și implementări ale acestora bazate pe metode de gradient (pasul descendent, pasul descendent relaxat, gradientul conjugat, gradientul conjugat scalat), care utilizează numai informațiile oferite de funcția de minimizat și gradientul ei. Adicional, biblioteca conține și modele specifice diverselor domenii de utilizare (energie, chimie, mecanică, economice - micro și macro, ecologice, mediu). Ideea este ca aditional soluției clasice, să se includă aceste modele ca expresia lor într-un limbaj de programare matematică (GAMS). În ceea ce privește implementarea tehnologiei GridMOSI, în sensul realizării comunicației dintre client și server cu scopul de a rezolva de la distanță probleme de optimizare, pentru început va fi abordată clasa problemelor de programare liniară, care este ceva mai simplă. Pentru aceasta se va încerca utilizarea tehnicilor dezvoltate în proiectul NEOS și Condor [18].

#### 4. Modelare numerică complexă și optimizare MDO în aplicații industriale

Oferta GridMOSI pentru acest domeniu constă într-un pachet de aplicații de optimizare multidimensională/multicriterială ce încorporează cele mai recente dezvoltări realizate la INCAS în probleme de aerodinamică aplicată. Acest pachet va permite abordarea unor probleme complexe și se bazează pe instrumente software și capacitate de calcul înglobate în MFCC (Modern Field Code Cluster). Rolul MFCC este de a oferi soluții (la nivel de solver) în vederea calculării valorilor funcției cost, care, în sens generalizat, caracterizează nivelul de performanță și calitatea procesului de design. Această metodă de cuantificare a funcției cost și a evaluării restricțiilor presupune în general utilizarea unor instrumente software de calcul de înaltă performanță, coduri numerice de tipul analizei structurale cu element finit, analiza curgerii tip CFD (Computational Fluid Dynamics) și, în general, presupune timpi foarte îndelungați de calcul și capacitate enormă de stocare.

Ideea implementării unor algoritmi și tehnici de optimizare în forma unor servicii Grid are la bază eforturile INCAS de transferare către servicii bazate pe tehnologie Grid a tehnicilor specifice de proiectare și optimizare în inginerie (aeronautică). Codurile de optimizare, indiferent de limbajul de programare în care sunt scrise sau mașinile pe care lucrează pot fi încapsulate în servicii Grid standard și pot fi făcute universal accesibile. Astfel devine posibil să se utilizeze într-un sistem integrat de proiectare un număr variabil de tehnici de optimizare sau, dimpotrivă, să se utilizeze o singură metodă de optimizare pentru un set de probleme de proiectare.

**Optimizer-ul bazat pe tehnologie Grid** își propune să utilizeze resurse software distribuite pentru calculul sensibilităților, în condițiile în care parametrii de bază calculați reprezintă soluții numerice ce au la bază programe complexe de analiză și simulare numerică. Prin acest tip de abordare procesul de optimizare numerică se transformă într-un meta-modul independent ce poate fi accesat din diferite limbaje de programare, utilizând variante distincte de middleware și în mod total transparent față de problemele de implementare. Întreg efortul este canalizat în ideea exploatării cu maximă eficiență a relației Optimizer – Analyzer, în condițiile în care Analyzer-ul reprezintă ponderea covârșitoare în timpul de calcul, atât ca volum de informație, cât mai ales ca timp de execuție.

Din punct de vedere tehnic, implementarea are la bază modelul serviciilor tranzitive OGSI - Open Grid Services Infrastructure. Prin migrarea în mediul Grid se are în vedere rescrierea algoritmilor de optimizare existenți și definirea unor șabloane pentru funcții cost specifice aplicației/utilizatorului.

**Cluster-ul aplicativ pentru analiză complexă** este conceput ca nucleu de bază pentru potențialul de

simulare existent la nivelul INCAS. Ideea unui cluster de resurse de simulare, pentru largă utilizare este bine primită în comunitatea științifică, în principal pentru domeniile CFD (Computational Fluid Dynamics) și analiza structurală tip FEM (Finite Element Analysis), precum și cuplaje CFD-FEM cu aplicații în probleme de fluturare, interacțiune structură-fluid, crash etc.

Soluția propusă se bazează pe aplicații specifice problemelor de optimizare multidimensională/multicriterială, unde algoritmi de bază și calculul sensibilităților se fac utilizând resurse din Grid sub formă de solvere modularizate. Modulele respective sunt la rândul lor aplicații/module gridificate. INCAS utilizează tehnologie Grid în ideea de a dezvolta o nouă versiune pentru MFCC (Modern Field Code Cluster), în condițiile integrării de capacități specifice calculului de înaltă performanță și comunicație rapidă. Clusterul MFCC este implementat pe o arhitectură hibridă, în special utilizând un cluster în arhitectura Beowulf existent la INCAS, dar și resurse delocalizate de tipul IBM-JUMP la FZJ-Germania.

## 5. Optimizare bazată pe algoritmi evolutivi

Algoritmii evolutivi, metode meta-euristice inspirate de evoluția în natură, sunt aplicate cu succes în rezolvarea aproximativă a problemelor de optimizare. Principalul avantaj îl reprezintă faptul că nu necesită satisfacerea unor precondiții de către problema de rezolvat și pot fi utilizați atât pentru optimizare globală cât și pentru optimizare multimodală și multicriterială. Sfera aplicațiilor algoritmilor evolutivi cuprinde probleme de planificare, design, simulare, identificare, control, data mining etc. din diverse domenii (inginerie, economie, biologie etc.). Dezavantajele sunt legate de faptul că sunt metode intensiv computaționale necesitând resurse semnificative atât din punct de vedere al spațiului de memorie utilizat datorită lucrului cu populații, dar mai ales în raport cu timpul de execuție (metode probabiliste cu convergență lentă).

Contribuția echipei de la Universitatea de Vest din Timișoara la oferta de soluții GridMOSI are la bază experiența în arhitecturi Grid [19], calcul paralel [20-22], calcul evolutiv [23]. Principalele rezultate obținute în domeniul calculului evolutiv se referă la: dezvoltarea unei metode de adaptare a parametrilor de control ai unui algoritm evolutiv de tip „differential evolution” și analiza aplicabilității acestuia pentru probleme de optimizare uni-criterială, multi-criterială și multi-modală, precum și dezvoltarea unei strategii de paralelizare a algoritmilor evolutivi bazată pe modelul sub-populațiilor și analiza eficienței sale în optimizarea uni și multi-criterială.

În ultimii ani au fost propuse diferite modele de paralelizare a algoritmilor evolutivi: master-slave, insular, celular. Alegerea adecvată a unui model de paralelizare depinde de particularitățile problemelor de rezolvat și ale arhitecturii pe care se va realiza implementarea. Cel mai potrivit pentru extindere în mediul Grid este modelul insular, caracterizat prin divizarea populației în mai multe (sub)populații la nivelul cărora se aplică un algoritm evolutiv de optimizare multicriterială. Ca exemple de implementări pe grid ale algoritmilor evolutivi pot fi menționate EVOLVE/G [24] și GOGA [25]. Dintre bibliotecile open-source destinate implementării distribuite a algoritmilor evolutivi pot fi amintite DREAM (Distributed Resource Evolutionary Algorithm Machine, <http://www.dcs.napier.ac.uk/~benp/dream/dream.htm>) și ParadisEO (PARAllel and DIStributed Evolving Objects, <http://www.lifl.fr/~cahon/paradisEO/index.html>).

**DEMO/G-Distributed Evolutionary Multiobjective Optimization on Grid.** Într-o primă etapă vor fi implementați algoritmi ce folosesc selecția bazată pe sortarea elementelor populației în funcție de rangul de nedominare combinată cu un mecanism de evitare a aglomerării elementelor în cadrul frontului Pareto (mecanism de tip “crowding”): NSGA-II - Non-dominated Sorting Genetic Algorithm [26], PDE – Pareto Differential Evolution [27].

În ceea ce privește modul de comunicare între (sub)populații, pe baza unei analize preliminare [28], s-a optat pentru implementarea variantelor de comunicare bazate pe topologiile cu conectare totală și înel, precum și pe selecția aleatoare, respectiv elitistă a emigranților. În cazul soluției GridMOSI, bazată pe extinderea implementării paralele pe Grid, se propune o strategie de comunicare rară și asincronă, prin care aplicația aferentă fiecărei locații din Grid (și care are în sarcină evoluția mai multor (sub)populații) trimite periodic (însă nu foarte frecvent) mesaje către toate celelalte locații și își analizează periodic buffer-ul de mesaje.

Domeniul general de aplicabilitate vizat de această soluție îl constituie rezolvarea problemelor de optimizare multicriterială fără restricții. În particular, va putea fi utilizată pentru rezolvarea de probleme de optimizare multicriterială ce apar în tehnică sau în probleme de grupare a datelor cu aplicabilitate concretă în segmentarea imaginilor, analiza datelor în genomică etc.

Interesul pentru gridificare este generat atât de paralelizarea, cât și de distribuirea în mediul Grid a taskurilor implicate în rezolvarea unei probleme de optimizare multicriterială, având ca motivație principală reducerea costului computațional al implementării.

## 6. Planificarea taskurilor în Grid

Problema balansării sarcinilor ocupă un rol esențial în domeniul Grid, atunci când se dorește folosirea optimă a resurselor disponibile. Unul dintre dezavantajele planificatoarelor centralizate este lipsa toleranței la defecte. La rândul lor, abordările descentralizate sunt focalizate pe partiționarea setului de taskuri în subseturi și rularea algoritmului pe fiecare dintre acestea. Multe dintre implementările de planificatoare privesc problema alocării prin prisma sistemelor multi-agent [30, 31], în care cooperarea și coordonarea tind să sporească semnificativ overloadul în sistem.

Soluția de **planificator distribuit bazat pe algoritmi genetici**, propusă de UPB este construită ca o aplicație Grid middleware, având rolul de a planifica grupuri de task-uri care trebuie executate pe un sistem distribuit (site Grid sau cluster), folosind pentru rularea efectivă a sarcinilor sisteme de execuție distribuită deja implementate, precum OpenPBS, Ganglia sau Condor. Utilizarea algoritmilor genetici reprezintă o abordare larg utilizată pentru probleme de planificare [29]. Modelul propus va putea rula într-o distribuție Grid cu cerințe minimale, în condiții real-time, monitorizând starea infrastructurii Grid și actualizând dinamic condițiile de execuție a task-urilor alocate.

În acest sens, sistemul este realizat sub forma unui serviciu web, oferit utilizatorilor care doresc să execute anumite grupuri de sarcini la diferite momente de timp. Se consideră că fiecare calculator din Grid este implicat în grupul de planificare (primește cererile de execuție task-uri, rulează algoritmul genetic și întoarce cea mai bună planificare obținută conform algoritmului de scheduling) și/sau grupul de execuție (execută task-urile care au fost planificate anterior și care le-au fost asignate ca urmare a planificării optime obținute de algoritmul). La nivelul grupului de planificare, planificatorul presupune existența unor noduri Master și a unor noduri Worker. Între aceștia există o comunicare punct-la-punct. Nodurile Master preiau cereri de planificare de task-uri provenite de la utilizatori, sub forma unui fișier care conține descrierea XML a task-urilor. Grupul de task-uri este trimis nodurilor Worker și plasat într-o coadă pe fiecare dintre aceștia. Algoritmul genetic începe atunci când coada de task-uri nu este goală și fie s-a completat un cromozom, fie a trecut o perioadă prestabilită de așteptare. Modelul realizat oferă o scalabilitate ridicată, în care un Worker poate apărea sau poate dispărea în orice moment de timp, fără ca acest fapt să afecteze funcționalitatea planificatorului. Caracterul distribuit al sistemului realizat oferă, pe lângă scalabilitate, și atribute de performanță ridicată prin creșterea vitezei de execuție a algoritmului genetic, precum și o bună toleranță la defecte.

Folosirea informațiilor de monitorizare privind calculatoarele din Grupul de execuție obținute cu ajutorul serviciului de MonALISA [32], duce la posibilitatea utilizării calculatoarelor disponibile atât pentru planificare cât și pentru execuție. Abordarea distribuită duce la scăderea numărului de generații în care algoritmul converge. În plus, sunt rezolvate astfel probleme de scalabilitate și de toleranță la defecte.

## 7. Algoritmi criptografici și criptanalitici

În abordarea clasică, algoritmi criptografici și criptanalitici sunt foarte bine studiați atât din punct de vedere teoretic cât și din punct de vedere practic. Există numeroase implementări de referință ale acestora care sunt disponibile în domeniul public, printre care și unele implementări care aparțin autorilor propunerii (Cryptolib, Cryptos, Crypto PRO, module Prolog etc.) [33-35]. Tot mai mulți cercetători își îndreaptă atenția spre elaborarea unor scheme criptografice de înaltă performanță care să beneficieze la maxim de actualele resurse hardware, între care un loc aparte îl ocupă tehnologiile de tip Grid (ex: [36]). Studiile recente arată creșteri semnificative în performanță pentru execuția în mediu Grid (ex: Alchemi, Universitatea din Melbourne) a unor algoritmi criptografici consacrați (ex: DES, Blowfish, RC4), dar și unele probleme de overhead, ceea ce indică necesitatea de a aprofunda în continuare studiul acestor metode, precum și necesitatea descoperirii unor metode inovatoare care să se adapteze mai bine mediului Grid.

În cadrul acestei soluții GridMOSI sunt abordate următoarele clase de algoritmi: algoritmi de criptare cu cheie secretă, de tip bloc (AES, RC6, etc.); algoritmi de criptare cu cheie secretă, de tip stream (RC4, A5/1, A5/2, etc.); algoritmi de criptare cu cheie publică (RSA, etc.); funcții de hashing (SHA 256, SHA 384, SHA 512, etc.); algoritmi de generare de secvențe de numere (pseudo)aleatoare (bateria NIST, Mersenne Twister, Yarrow etc.); algoritmi de testare a secvențelor de numere aleatoare (bateria de teste NIST și Marsaglia); algoritmi criptanalitici (criptanaliza clasică (transpoziție, substituție), factorizare).

Pentru obținerea unei creșteri de performanță se folosesc două abordări: una bazată pe MPICH-G2 și una de tip SIMD. Principala cerință de performanță care se impune aici este dată de viteza de comunicație în rețea, care în etapa curentă se preconizează a fi de 1Gbps. Pentru accesarea datelor necesare rulării algoritmilor sunt evaluate comparativ trei soluții: *input sandbox*, *gsiftp* și *director partajat* la nivelul tuturor nodurilor de lucru.

## 8. Concluzii

Proiectul GridMOSI are un ridicat nivel de complexitate sub aspect profesional și managerial. Din punct de vedere profesional, complexitatea este generată, în primul rând, de caracterul interdisciplinar al proiectului. Prin obiectivele propuse, proiectul GridMOSI este aliniat la principalele orientări actuale în dezvoltarea Grid pe plan regional și european: compatibilizarea și integrarea infrastructurilor, dezvoltarea și diversificarea ofertei de aplicații, facilitarea accesului utilizatorilor, armonizarea prin proiecte a acțiunilor și resurselor dedicate domeniului.

Lucrarea a fost dedicată componentei aplicaționale a proiectului, care poate prezenta interes pentru o colectivitate largă de cercetători, cadre didactice și utilizatori specializați în problematica soluțiilor MOSI promovate în cadrul proiectului.

## Bibliografie

1. \* \* \*: EC, DG INFSO. Changing the way research is done. The user perspective on eInfrastructure. Bruxelles, decembrie 2003, p. 3.
2. \* \* \*: ANCS. Pilot project for the implementation of the national Grid infrastructure – Strategic Plan. București, februarie 2006, p. 10.
3. **FOSTER, I., C. KESSELMAN, S. TUECKE**: The Anatomy of the Grid Enabling Scalable Virtual Organizations. *International J. Supercomputer Applications*, 15(3), 2001.
4. **SIMA, V.**: SLICOT-based advanced automatic control computations. *Advances in Automatic Control* (M. Voicu, Ed.), *Kluwer Academic Publishers*, Boston/Dordrecht/ London, 2003, pp. 337-350, ISBN 1-4020-7607-X.
5. **VAN HUFFEL, S., V. SIMA, A. VARGA, S. HAMMARLING, F. DELEBECQUE**: Development of high performance numerical software for control. *IEEE Control Systems Magazine*, 24(1): 60-76, Feb. 2004.
6. **SIMA, V.**: Identification of a steel subframe flexible structure using SLICOT system identification toolbox. În CD-ROM Proc. of the 11th Mediterranean Conference on Control and Automation MED'03, June 18-20 2003, Rhodes, Greece, 2003. Invited Session IV01 "Computational Toolboxes În Control Design", June 19 2003, Paper IV01-07, 6 pages.
7. **SIMA, V.**: Efficient Data Processing for Subspace-based Multivariable System Identification. În: Proc. of The IFAC Workshop on Adaptation and Learning in Control and Signal Processing (ALCOSP 2004), August 30-September 1, 2004, Yokohama, Japan, pp. 871-876.
8. **SIMA, V., D.M. SIMA, AND S. VAN HUFFEL**: High-performance numerical algorithms and software for subspace-based linear multivariable system identification. *J. Comput. Appl. Math.*, 170(2):371-397, 2004.
9. **SIMA, V.**: Performance investigation of SLICOT Wiener Systems Identification Toolbox. În: Proc. of the 13th IFAC Symposium on System Identification, SYSID 2003, Aug.27-29, 2003, Rotterdam, The Netherlands, pages 1345-1350, Omnipress.
10. **ANDREI, N.**: Sisteme și pachete de programe pentru programarea matematică. Editura Tehnică, București, 2002.
11. **ANDREI, N.**: Programarea Matematică Avansată. Teorie, Metode Computaționale, Aplicații. Editura Tehnică - București, 1999.
12. **ANDREI, N.**: Optimizare fără restricții. Metode de direcții conjugate. Editura MATRIXROM, București, 2000.
13. **ANDREI, N.**: Preliminary Computational Experience with ASLO an Advanced System for Linear Optimization. Technical Report No. AMOL-96-6, Research Institute for Informatics, Bucharest, August 22, 1996.
14. **MURTAGH, B. A., M.A. SAUNDERS**: MINOS 5.5 User's Guide, Report SOL 83-20R, Systems Optimization Laboratory, Stanford University (revised July 1998).
15. **BROOKE, A., D. KENDRICK, A. MEERAUS, R. RAMAN**. GAMS: A user's Guide. GAMS Development Corporation, December 1998.
16. **FOURER, R., D. M. GAY, B. W. KERNIGHAN**: AMPL. A Modeling Language for Mathematical Programming, Duxbury Press / Brooks/Cole Publishing Company, 1993, 351 + xvi p., ISBN 0-534-50983-5.

17. **ANDREI, N., G. BORCAN:** ALLO - limbaj algebric pentru optimizare liniară. *Revista Română de Informatică și Automatică*, vol. 8, nr. 3, 1998, pp. 55-67.
18. **FERRIS, M. C., M.P. MESNIER, J.J. MORÉ:** NEOS and Condor. Solving Optimization Problems over the Internet. *ACM TOMS*, vol.26, no.1, 2000, pp. 1-18.
19. **PETCU, D., D. TEPENEU, M. PAPRZYCKI, T. IDA:** Symbolic Computations on Grids, invited chapter 6 în the book "Engineering the Grid: status and perspective", eds. Beniamino di Martino, Jack Dongarra, Adolfo Hoisie, Laurence Yang, and Hans Zima, ISBN: 1-58883-038-1, 2006.
20. **PETCU, D.:** Experiments with an ODE Solver on a Multiprocessor System, *Computers & Mathematics with Applications* vol. 42, no. 8-9, Oct.-Nov. (2001), Pergamon-Elsevier Science, UK, pp. 1189-1199.
21. **PETCU, D.:** The Performance of Parallel Iterative Solvers, *Computers & Mathematics with Applications* vol. 50 (2005), Pergamon-Elsevier Science, UK, pp. 1179-1189.
22. **BONCHIS, C., G. CIOBANU, C. IZBASA, D. PETCU:** A Web-Based P Systems Simulator and Its Parallelization, C.S.Calude et al. (Eds.): UC 2005, LNCS 3699, pp. 5869, 2005.
23. **ZAHARIE, I., D. ZAHARIE:** Evolutionary optimization of molecular clusters. *Proc. of 4th Conference on Isotopic and Molecular Processes, Cluj-Napoca*, sept. 22-24, 2005.
24. **YUSUKE TANIMURA, TOMOYUKI HIROYASU, et al.:** The System for Evolutionary Computing on the Computational Grid, *IATED 14th International Conference on Parallel and Distributed Computing and Systems*, 2002, pp.39-44.
25. **IMADE, H., R. MORISHITA, I. ONO, N. ONO, M. OKAMOTO:** A Grid-Oriented Genetic Algorithm Framework for Bioinformatics. *New Generation Computing*, Vol.2, (2004) pp. 177 – 186.
26. **DEB, K., AMRIT PRATAP, SAMEER AGARWAL, T. MEYARIVAN:** A Fast and Elitist Multi-Objective Genetic Algorithm-NSGA-II, *IEEE Trans. On Evolutionary Computation*, vol. 6, pp. 182-197, 2002.
27. **MADAVAN, N.:** Multiobjective Optimization Using a Pareto Differential Evolution Approach. In Fogel et al., editors, *Proc. of the 2002 Congress of Evolutionary Computation*, vol. 1, 2002, pp. 1145-1150.
28. **ZAHARIE, D., D. PETCU:** Communication Strategies in Distributed Evolutionary Algorithms for Multi-objective Optimization". În *Proc. 7th International Conference on Technical Informatics - CONTI2006*, eds. O. Prostean et al, Ed. Politehnica, Timisoara, 2006, vol. 1: Automation and Applied Informatics, pp 151-156.
29. **PAGE, A. J., T.J. NAUGHTON:** Dynamic task scheduling using genetic algorithms for heterogeneous distributed computing". În: *Proc. of the 19th International Parallel & Distributed Processing Symposium*, , Denver, Colorado, USA, April 2005, pp. 189a.1-189a.8.
30. **CAO, J., D.P. SPOONER, S. A. JARVIS, S. SAINI, G.R. NUDD:** Grid load balancing using intelligent agent". În: *Future Generation Computer Systems special issue on Intelligent Grid Environments: Principles and Applications*, 2004.
31. **WEICHHART, G., M. AFFENZELLER, A. REITBAUER, S. WAGNER:** Modelling of an Agent-Based Schedule Optimisation System". În: *Proc. of the IMS International Forum*, 2004.
32. **NEWMAN, H. B., I.C. LEGRAND, P. GALVEZ, R. VOICU, C. CIRSTOIU:** MonALISA: A Distributed Monitoring Service. *CHEP 2003*, La Jolla, California, March 2003.
33. **PUSZTAI, K., T. MUREȘAN, R. POTOLEA, A. SUCIU:** Constraint Solving and Complex Computations în Engineering – Modern Approaches to Constraint Solving and Optimization (Final Report). *Research Report*, Technical University, Cluj-Napoca, DaimlerChrysler, April 7, 2001.
34. **SUCIU, A., K. PUSZTAI, C. IONESCU:** Design and Implementation of a Cryptographic Tool. *Proc. of CONTI 2002*, Timisoara, October 18-19, 2002, pp.47-52.
35. **SUCIU, A.:** Integrarea programării logice în limbaje și sisteme multiparadigmă. Editura U.T.Pres, Cluj-Napoca, 2005, ISBN 973-662-176-6.
36. **SETIAWAN, A., D. ADIUTAMA, J. LIMAN, A. LUTHER, R. BUYYA:** GridCrypt: High Performance Symmetric Key Cryptography using Enterprise Grids. *GridBus Project*, 2005 (<http://www.gridbus.org/papers/gridcrypt.pdf#search=%22GridCrypt%22>).