

Universitatea „Ștefan cel Mare” Suceava
Facultatea de Inginerie Electrică și Știința Calculatoarelor
Specializare Știința și ingineria calculatoarelor

USVScheduleGA: O abordare evolutiva pentru problema de programare universitara

Profesor:
RUSU Ionela

Masterand:
AYDOGAN Mert
LIONTE Denis

Rezumat

Această documentație prezintă proiectul **USVScheduleGA**, o soluție evolutivă destinată rezolvării Problemei de Programare Universitară (University Timetabling Problem – UTP), recunoscută în literatura de specialitate ca o problemă combinatorială NP-dura (Carter & Laporte, 1997). Obiectivul principal este de a atribui cursurilor, seminarelor și laboratoarelor, intervale orare și a unor săli adecvate, respectând atât constrângerile dure (capacitate, disponibilitate, conflicte profesori și grupe), cât și constrângerile flexibile asociate preferințelor actorilor academici. Într-un context real, generarea unui orar fezabil este dificilă, deoarece numărul total de combinații posibile crește exponențial odată cu mărimea instituției, iar soluțiile manuale sunt adesea consumatoare de timp și predispose la erori (Schaerf, 1999).

Proiectul propune un algoritm genetic (GA) implementat în Java 21 și orchestrat printr-o aplicație Spring Boot, utilizând un model de fitness pentru evaluarea calității soluțiilor. Prin explorarea populațiilor de orare și aplicarea operatorilor evolutivi (selecție, încrucișare, mutație și elitism), sistemul convertește spre configurații fezabile care reduc conflictele și maximizează respectarea preferințelor. Rezultatele experimentale indică faptul că metoda evolutivă poate obține soluții satisfăcătoare într-un interval de timp rezonabil, demonstrând astfel utilitatea practică a abordării în context universitar (Babaei et al., 2015).

Introducere

Proiectul **USVScheduleGA** a fost dezvoltat în context academic și tehnic pentru a investiga aplicabilitatea algoritmilor genetici asupra unei probleme reale cu impact administrativ major: generarea orarelor universitare. Instituțiile de învățământ superior se confruntă anual cu procesul dificil de planificare a activităților academice (Schaerf, 1999), proces care implică resurse limitate, un număr ridicat de entități și multiple preferințe contradictorii. Metodele tradiționale bazate pe planificare manuală sau pe reguli euristice simple conduc, în general, la soluții suboptimale și dificil de ajustat în timp (Lewis, 2008).

USVScheduleGA propune utilizarea unui **algoritm genetic (GA)** pentru explorarea spațiului mare de soluții și pentru identificarea unor orare fezabile într-un mod automatizat și

transparent. Sistemul este implementat în Java 21 și se bazează pe Spring Boot pentru orchestrarea serviciilor, oferind o arhitectură monolitică robustă și ușor extensibilă. Alegerea unui GA se justifică prin capacitatea sa de a mixa soluții parțiale bune, de a evita blocarea în optime locale și de a optimiza simultan atât constrângerile dure, cât și pe cele soft. În plus, proiectul servește ca studiu de caz pentru înțelegerea modului în care tehnicile evolutive pot fi integrate în sisteme software moderne, adresând probleme administrative reale și reducând substanțial efortul uman implicat (Goldberg, 1994; Sastry et al., 2013).

Formularea problemei

Problema de programare este formulată ca o mapare între un set de evenimente E și perechi (slot temporar, sală). Fiecare eveniment are cerințe (număr de studenți, tip de sală) și trebuie plasat într-un slot în care nu intra în conflict cu alte evenimente. Constrângerile dure includ:

- **Timpul maxim:** o grupă nu poate avea mai mult de 8 ore într-o singură zi;
- **Incompatibilitate:** anumite evenimente nu pot avea loc simultan (de exemplu același profesor sau aceeași grupă de studenți);
- **Disponibilitate:** evenimentul trebuie să se potrivească în ferestrele permise.

Scopul optimizării este de a genera un orar care respectă toate aceste constrângeri și minimizează penalitățile pentru preferințe nesatisfacute, cum ar fi orele matinale sau zilele aglomerate.

Metodologie

Algoritmul genetic folosit în USVScheduleGA urmează etapele standard:

- **Codarea cromozomului:** fiecare cromozom reprezintă un orar complet; genele sunt atribuiri de eveniment \rightarrow (slot, sală);
- **Populația inițială:** se creează o mulțime de ore generate aleator respectând în mare măsură constrângerile;
- **Funcția de fitness:** evaluează cât de bun este un orar, penalizând încălcările și recompensând respectarea preferințelor;

- **Selectia:** cromozomii cu fitness mai ridicat sunt selectati pentru reproducere. Se aplica metoda turneu pentru diversitate;
- **Incrucisare (crossover):** fragmente din doi parinti sunt combinate pentru a produce copii, pastrand structuri de orar bune;
- **Mutare:** se modifica aleator atribuirea unui eveniment pentru a explora noi configuratii;
- **Elitism:** cele mai bune solutii sunt pastrate de la o generatie la alta pentru a nu pierde calitatea.

Algoritmul ruleaza pana la atingerea unui numar de generatii sau pana cand imbunatatirile devin nesemnificative.

Evaluare si rezultate

USVScheduleGA a fost testat pe date reale de la Universitatea Stefan cel Mare din Suceava. Rezultatele au aratat ca:

- Algoritmul genetic produce orare fezabile care respecta toate constrangerile dure;
- Fitness-ul final se imbunatateste semnificativ pe parcursul generatiilor;
- Timpul de calcul este acceptabil pentru utilizare practica, iar parametrii (marimea populatiei, rata de crossover, rata de mutatie) pot fi ajustati pentru a echilibra calitatea si performanta.

Aceste rezultate demonstreaza viabilitatea abordarii evolutive pentru problema programarii universitare.

Comparatie cu alti algoritmi

Problema de programare a orarului a fost abordata in literatura folosind diverse tehnici algoritmice. Comparativ cu algoritmul genetic din USVScheduleGA, alte abordari includ:

- **Algoritmi heuristici si greedy:** Metode simple care construiesc orarul pas cu pas pe baza unor reguli fixe. Acestea sunt rapide, dar pot ramane blocate in solutii suboptime si nu exploreaza spatiul complet de solutii. (Babaei 2015)
- **Cautare tabuu si recocire simulata:** Tehnici metaeuristice care pornesc de la o solutie initiala si o imbunatatesc prin schimbari locale, acceptand ocazional deteriorarea pentru

a evita blocajele. Acestea pot oferi rezultate bune, dar necesita reglarea atenta a parametrilor. (Babaei 2015)

- **Programare liniara si satisfacerea constrangerilor:** Formuleaza problema ca un set de ecuatii sau restrictii si utilizeaza solve specializate. Aceste metode pot garanta optimalitatea pentru exemple mici, dar devin ineficiente pentru probleme mari. (Babaei 2015)

Algoritmul genetic ofera un compromis intre calitate si flexibilitate, permitand explorarea unui spatiu larg de solutii si ajustarea parametrilor pentru a obtine rezultate mai bune decat metodele simple, cu un timp de calcul rezonabil comparativ cu tehnici mai sofisticate pentru aplicarea tehnicilor evolutive, cat si ca o solutie potentiala pentru institutii care doresc sa isi automatizeze procesul de planificare a orarului.

Concluzii

USVScheduleGA ofera o abordare evolutiva moderna pentru rezolvarea problemei NP-dure a programarii orelor din cadrul universitatii. Prin modelarea problemei ca un algoritm genetic, sistemul obtine orare care respecta constrangerile academice si preferintele participantilor. Proiectul serveste atat ca un studiu de caz.

Bibliografie

- Babaei, H., Karimpour, J., & Hadidi, A. (2015). A survey of approaches for university course timetabling problem. *Computers & Industrial Engineering*, 86, 43-59.
- Carter, M. W., & Laporte, G. (1997). Recent developments in practical course timetabling. 3-19.
- Goldberg, D. E. (1994). Genetic and evolutionary algorithms come of age. *Communications of the ACM*, 37(3), 113-120.
- Lewis, R. (2008). A survey of metaheuristic-based techniques for university timetabling problems. *OR spectrum*, 30(1), 167-190.
- Sastry, K., Goldberg, D. E., & Kendall, G. (2013). Genetic algorithms. In *Search methodologies: Introductory tutorials in optimization and decision support techniques* (pp. 93-117). Springer.
- Schaerf, A. (1999). A survey of automated timetabling. *Artificial intelligence review*, 13(2), 87-127.