

# Optimasi *Extended Genetic Algorithm* Dalam Memecahkan Masalah Penjadwalan Perkuliahan Dengan Strategi *Greedy*

Wawan Gunawan

Universitas Mercu Buana  
Jl. Kembangan No. 1 Jakarta, Indonesia

e-mail: wawan.gunawan@mercubuana.ac.id

---

Informasi Artikel      Diterima: 25-05-2021      Direvisi: 03-08-2021      Disetujui: 18-08-2021

---

## Abstrak

Penentuan jadwal merupakan proses yang cukup rumit karena melibatkan beberapa faktor determinan, bagi tingkat universitas penentuan jadwal mengajar dosen sangat ditentukan berdasarkan keahlian masing-masing dosen dalam mengampu mata kuliah dan juga ketersediaan waktu mengajar. Penjadwalan pada tingkat universitas akan dipengaruhi oleh jam mengajar, ruang kelas, dosen pengampu, mata kuliah, kelas. Pada penelitian ini kami mengusulkan penggunaan *extended genetic algorithm (EGA)* dalam penjadwalan dengan strategi *greedy* dan pembatasan jumlah mengajar dosen berdasarkan kebijakan universitas untuk masing-masing jenis dosen. Proses perhitungan berdasarkan algoritma dan juga kemudahan aplikasi telah sesuai atau belum. Hasil dari kuisioner penilaian yang dilakukan oleh 20 orang pengguna telah memberikan nilai 100% untuk kesesuaian proses perhitungan yaitu dari penilaian A – E, sedangkan untuk kenyamanan aplikasi dari point F – G adalah sebesar 92,5% yang menyatakan nyaman, dan 7,5% yang menyatakan kurang nyaman.

**Kata Kunci:** *Extended Genetic Algorithm, Greedy, penjadwalan*

## Abstract

*Determining the schedule is a fairly complicated process because it involves several determinant factors, for the university level determining the teaching schedule for lecturers is very much determined based on the expertise of each lecturer in teaching courses and also the availability of teaching time. Scheduling at the university level will be influenced by teaching hours, classrooms, lecturers, courses, classes. In this study we propose the use of the extended genetic algorithm (EGA) in scheduling with a greedy strategy and limiting the number of lecturers teaching based on university policies for each type of lecturer. The calculation process is based on algorithms and also the ease of application is appropriate or not. The results of the assessment questionnaire conducted by 20 users have given a value of 100% for the suitability of the calculation process, namely from the A - E assessment, while for the convenience of application from point F - G is 92.5% which states comfortable, and 7.5% which states less comfortable.*

**Keywords:** *Extended Genetic Algorithm, Greedy, penjadwalan*

## 1. Pendahuluan

Penjadwalan merupakan suatu keharusan yang dilakukan oleh masing-masing individu dan juga tiap institusi, agar aktivitas pekerjaan yang dijalankan sesuai dengan proses bisnis yang telah dibuat. Melakukan penjadwalan merupakan suatu hal yang sangat serius bagi institusi agar dapat menyeimbangkan konflik manfaat diantara sesama *stakeholder* (Zhang et al., 2019), dalam hal ini adalah dosen. Penentuan jadwal merupakan proses yang cukup rumit karena melibatkan beberapa faktor determinan (Wijyaningrum & Mahmudy, 2016), bagi tingkat

universitas penentuan jadwal mengajar dosen sangat ditentukan berdasarkan keahlian masing-masing dosen dalam mengampu mata kuliah dan juga ketersediaan waktu mengajar dosen terkait.

Permasalahan yang terjadi akan terlihat ketika banyak dosen yang memiliki keahlian dan waktu tersedia yang sama untuk mengajar, sedangkan kuota yang tersedia memiliki keterbatasan. Pun sebaliknya ketika kuota mengajar yang tersedia melebihi dari ketersediaan dosen yang mengajar atau adanya jadwal yang bentrok (Toscany & Roestam, 2017)



karena terpengaruh dengan peraturan-peraturan yang berlaku (Mauluddin et al., 2017).

Penjadwalan yang dilakukan digunakan untuk mengoptimalkan (Mangkuto, 2017)(Hadi, 2017) seluruh keahlian dosen yaitu dengan meminimalkan waktu eksekusi dan memaksimalkan parelisasi dalam menentukan jadwal dosen (Akbari et al., 2017) serta mempercepat proses pembuatan jadwal (Mauluddin et al., 2017)(Ni Luh Gede Pivin et al., 2016) yang bernilai objektif (Rahmani Hosseinabadi et al., 2019), karena kompetensi akan berpengaruh positif terhadap kinerja (Gede et al., 2016).

Penelitian sebelumnya menginformasikan bahwa penjadwalan menggunakan *extended genetic algorithm (EGA)* dengan metode *interval-valued intuitionistic fuzzy entropy weight (IVIFEW)* yang mengintegrasikan operator migrasi dan pencarian lokal dapat memberikan solusi optimal yang lebih baik dibandingkan dengan algoritma dasar yaitu *hybrid genetic algorithm (HGA)*, *extended variable neighborhood search (EVNS)*, *two-stage biogeography-based optimization (TBBO) algorithm* karena telah mampu memecahkan masalah ketika menggunakan model *fuzzy mixed-integer linear programming (FMILP)* (Zhang et al., 2019)(Borisovsky et al., 2020)(Zhou et al., 2020). Selain itu pada penelitian lainpun menyatakan bahwa penggunaan algoritma genetik pun telah dilakukan dalam membantu mengoptimalkan penjadwalan (Nitisiri et al., 2019) kapal untuk meminimalkan (Feng et al., 2017) jarak masing-masing kapal, di mana terdapat beberapa pelabuhan yang akan dikunjungi oleh beberapa kapal.

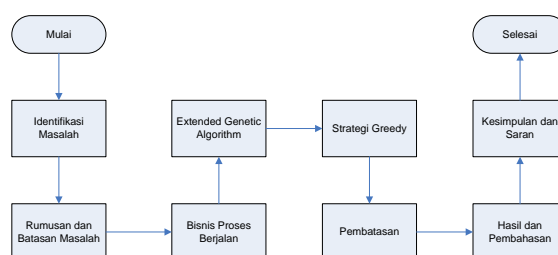
Dari percobaan komputasi yang telah dilakukan menghasilkan parameter optimal (Zan et al., 2020) dengan jumlah *popsiz* sebanyak 30, jumlah generasi sebanyak 100, nilai *crossover rate* sebesar 0,3 dan mutasi nilai *rate* adalah 0,7 (Wijayaningrum & Mahmudy, 2016). Selain itu, algoritma genetica pun memiliki nilai yang tidak berubah mulai generasi ke-60 hingga generasi ke-300 dengan nilai *fitness* 216,51 (Manggolo et al., 2017). Untuk penjadwalan yang memiliki keteraturan, maka algoritma genetik yang diusulkan adalah *metaheuristic* karena merupakan yang paling efisien (Abreu et al., 2020) dan efektif untuk menyelesaikan *Flexible Job-shop Scheduling Problem (FJSP)* (Pezzella et al., 2008). Selanjutnya untuk melakukan verifikasi algoritma dapat menggunakan metode *Labeled Transition System (LTS)*, *NuSMV* dan *PAT* (Keshanchi et al., 2017).

Penjadwalan pada tingkat universitas akan dipengaruhi oleh jam mengajar, ruang

kelas, dosen pengampu (Toscany & Roestam, 2017), mata kuliah dan juga kelas (Mauluddin et al., 2017). Pada penelitian ini kami mengusulkan optimasi penjadwalan menggunakan *extended genetic algorithm (EGA)* dengan strategi *greedy* dan pembatasan jumlah mengajar dosen berdasarkan kebijakan universitas untuk masing-masing jenis dosen. Algoritma dengan prinsip *greedy* dapat memecahkan masalah dengan jalur terpendek untuk graf langsung dengan bobot non-negatif (Nurhaida et al., 2019).

## 2. Metode Penelitian

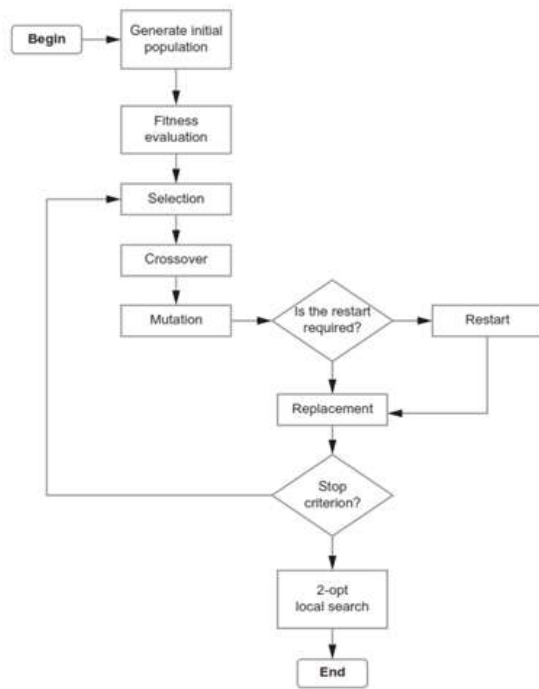
Penelitian ini akan melakukan proses pencocokan antara form pengisian kesediaan mengajar untuk 10 orang dosen dengan jadwal mengajar yang telah dibuat pada semester berjalan. Penggunaan algoritma genetica ini akan dibatasi pada penggunaan ruang kelas yang digunakan sebanyak 5 kelas, jumlah dosen sebanyak 10 orang dosen, jumlah mahasiswa sebanyak 150 orang mahasiswa, jumlah mata kuliah sebanyak 10 mata kuliah dan lokasi kampus yang akan digunakan berada di Meruya. Penggunaan Algoritma Genetik akan dilakukan penambahan proses pada saat melakukan *crossover* dengan menggunakan metode *greedy* sehingga akan ditemukan *Extended Genetic Algorithm*.



Gambar 1 Metode Penelitian

## 3. Hasil dan Pembahasan

Dari alur algoritma genetica, maka penulis menjabarkan persamaan dari proses tersebut dengan *Flowchart* pada gambar 2



Gambar 2 Flowchart Metode Algoritma Genetika

Dari alur algoritma tersebut, maka penulis menjabarkan persamaan sebagai berikut:

```

1. function nilaiFitness($target,$gen){
2.     $panjang_target = strlen($target);
3.     $s1=$target;
4.     $s2=$gen;
5.     $tot=0;
6.     for($a=0;$a<strlen($target);$a++){
7.         $s1=substr($target,$a,1);
8.         $s2=substr($gen,$a,1);
9.
10.        if($s1==$s2){
11.            $tot = $tot+1;
12.        }
13.    }
14.    $fitness = ($tot/$panjang_target)*100;
15.    return $fitness;
16. }
    
```

Gambar 3 Mencari Nilai Fitness

```

1. function representasiGenetik($target){
2.     $panjang_target = strlen($target);
3.     $gen = "";
4.     $characters = array_merge(range('A','Z'),
5.         range('a','z'), range('0','9'));
6.     $max = count($characters) - 1;
7.     for ($i = 0; $i < $panjang_target; $i++)
8.     {
9.         $rand = mt_rand(0, $max);
10.        $gen .= $characters[$rand];
11.    }
    
```

```

12.     return $gen;
13. }
    
```

Gambar Error! No text of specified style in document.4 Representasi Genetik

```

1. function cariPopulasi($besar_populasi,$target)
2. {
3.     for ($i=0;$i<$besar_populasi;$i++){
4.         $gene = representasiGenetik($target);
5.         $data[$i]['gen'] = $gene;
6.         $data[$i]['fitness'] = nilaiFitness($target,$gene);
7.     }
8.     return json_encode($data);
9. }
10. }
    
```

Gambar 5 Mencari Populasi

```

1. function selection($besar_populasi,$target){
2.
3.     $parent1="";
4.     $parent2="";
5.
6.     do {
7.         $urut = 0;
8.
9.
10.        $populasi = cariPopulasi($besar_populasi,$target);
11.
12.        $arr = json_decode($populasi, true);
13.
14.        foreach ($arr as $MK) {
15.
16.            if($MK['fitness']>0){
17.                $parent1 = $MK['fitness']." ".$MK['gen']."<br>";
18.                $parentGen1 = $MK['gen'];
19.                $parentFit1 = $MK['fitness'];
20.
21.                break;
22.            }
23.            $urut++;
24.        }
25.        unset($arr[$urut]);
26.        foreach ($arr as $MK) {
27.            //echo "B".$MK['fitness']." ".$MK['gen']."<br>";
28.            if($MK['fitness']>0){
29.                $parent2 = $MK['fitness']." ".$MK['gen']."<br>";
30.                $parentGen2 = $MK['gen'];
31.                $parentFit2 = $MK['fitness'];
    
```

```

32.         break;
33.     }
34. }
35.
36. } while (empty($parent1) or empty($parent2));
37.
38. $parent[0]['gen']=$parentGen1;
39. $parent[0]['fitness']=$parentFit1;
40. $parent[1]['gen']=$parentGen2;
41. $parent[1]['fitness']=$parentFit2;
42. return json_encode($parent);
43.
44.
45. }
    
```

Gambar 6 Selection Genetik

```

1. function crossOver($parentGenFitnes){
2.
3.     $parentGen1 = $parentGenFitnes[0]['gen'];
4.
5.     $parentGen2 = $parentGenFitnes[1]['gen'];
6.
7.     $child1 = substr($parentGen2,0,round(strlen($parentGen2)/2)).substr($parentGen1,round(strlen($parentGen2)/2),(strlen($parentGen1)-(round(strlen($parentGen2)/2))));
8.     $child2 = substr($parentGen1,0,round(strlen($parentGen1)/2)).substr($parentGen2,round(strlen($parentGen1)/2),(strlen($parentGen2)-(round(strlen($parentGen1)/2))));
9.
10.    $child[0]['child']=$child1;
11.    $child[1]['child']=$child2;
12.    return json_encode($child);
13. }
    
```

Gambar 7 Crossover Genetik

```

1. function mutasi($child,$laju_mutasi){
2.     $hasilMutan="";
3.     $min = 0;
4.     $max = 1;
5.     for ($a=0;$a<strlen($child);$a++){
6.         $nilaiRandom = mt_rand($min*10, $max*10) / 10;
7.
8.         $characters = array_merge(range('A','Z'), range('a','z'), range('0','9')); //, range('!', '@'), range('[', '_'), range('{', '}')
9.         $maxi = count($characters) - 1;
10.        $rand = mt_rand(1, $maxi);
11.
12.        if ($nilaiRandom<=$laju_mutasi){
13.            $hasilMutan .= $characters[$rand]
14.        } else {
15.            $hasilMutan .= substr($child,$a,1
    );
    
```

```

16.     }
17.
18. }
19.
20. return $hasilMutan;
21. }
    
```

Gambar 8 Mutasi Genetik

## Software Architecture

Aplikasi ini dirancang dengan menggunakan bahasa pemrograman berbasis objek oriented. Beberapa diagram yang akan dimodelkan menggambarkan rancangan sistem ini, yang terdiri dari *use case diagram*, dan *activity diagram*.

### 1. Diagram Use Case

*Use Case* merupakan pemodelan untuk kegiatan sistem informasi yang akan dibuat.



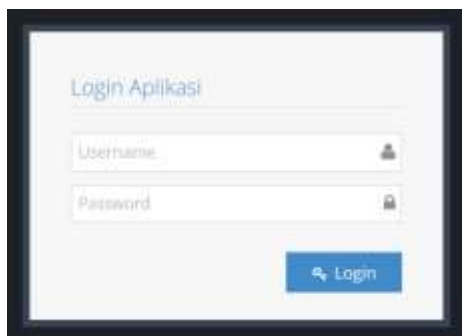
Gambar 9 Use Case Diagram

## Implementasi

Setelah pembuatan aplikasi selesai maka selanjutnya dilakukan implementasi.

### 1. Tampilan Login

Halaman ini merupakan halaman utama dari Aplikasi Penjadwalan menggunakan Algoritma Genetika dengan metode greedy seperti pada gambar 10.



Gambar 10 Tampilan Login

## 2. Tampilan Data Fakultas

Halaman ini akan tampil ketika kita ingin melihat data fakultas apa saja yang ada pada Universitas seperti terlihat pada gambar 11.



Gambar 11 Tampilan Data Fakultas

## 3. Tampilan Program Studi

Di menu ini menampilkan proses data Program Studi berdasarkan Fakultas yang ada pada universitas seperti pada gambar 12.



Gambar 12 Tampilan Data Program Studi

## 4. Tampilan Data Mata Kuliah

Halaman ini akan tampil seluruh data mata kuliah pada tiap program studi dan juga pada tiap fakultas yang nantinya akan menjadi rujukan untuk kesediaan mengajar dosen dan jadwal mengajar seperti pada gambar 13



Gambar 13 Tampilan Data Mata Kuliah

## 5. Tampilan Data Dosen

Halaman ini akan tampil seluruh dosen-dosen pada tiap program studi dan juga pada tiap fakultas yang nantinya akan menjadi rujukan untuk kesediaan mengajar dosen dan jadwal mengajar seperti pada gambar 14



Gambar 14 Tampilan Data Dosen

Selanjutnya berdasarkan hasil pengujian terhadap pengguna melalui kuesioner yang diberikan dengan pertanyaan seperti pada tabel 1 dengan 20 koresponden didapatkan hasil seperti pada tabel 2.

Tabel 1. Daftar penilaian kuesioner

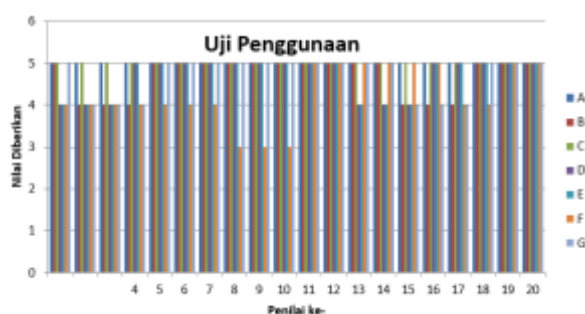
Kode	Keterangan
A	Kejelasan input yang digunakan
B	Implementasi Algoritma
C	Perhitungan yang dihasilkan
D	Lamanya proses perhitungan algoritma
E	Isi materi sesuai indikator
F	Aplikasi yang dihasilkan
G	Bahasa yang digunakan mudah dipahami

Sehingga didapatkan hasil bahwa sebanyak 92,5% yang menyatakan nyaman dengan aplikasi yang diberikan dan mampu membantu hasil dari penjadwalan.

Tabel 2. Data korespondensi

Penilai Kode	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
B	5	4	4	4	5	5	5	5	5	5	5	5	5	5	4	4	4	5	5	5
C	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
D	4	4	4	5	5	5	5	5	5	5	5	5	4	4	4	5	5	5	5	5
E	4	4	4	5	5	5	5	5	5	5	5	5	4	4	4	5	5	5	5	5
F	4	4	4	4	4	4	4	3	3	3	5	5	5	5	5	5	4	4	5	5
G	5	4	4	4	5	5	5	5	5	5	5	5	5	5	4	4	4	5	5	5

Dari hasil responden dapat ditampilkan dalam bentuk grafik seperti pada gambar 15.



Gambar 15 Grafik penilaian aplikasi

#### 4. Kesimpulan

Proses perhitungan berdasarkan algoritma dan juga kemudahan aplikasi telah sesuai atau belum. Hasil dari kuisioner penilaian yang dilakukan oleh 20 orang pengguna telah memberikan nilai 100% untuk kesesuaian proses perhitungan yaitu dari penilaian A – E, sedangkan untuk kenyamanan aplikasi dari point F – G adalah sebesar 92,5% yang menyatakan nyaman, dan 7,5% yang menyatakan kurang nyaman.

#### Referensi

- Abreu, L. R., Cunha, J. O., Prata, B. A., & Framinan, J. M. (2020). A genetic algorithm for scheduling open shops with sequence-dependent setup times. *Computers and Operations Research*, 113, 104793. <https://doi.org/10.1016/j.cor.2019.104793>
- Akbari, M., Rashidi, H., & Alizadeh, S. H. (2017). An enhanced genetic algorithm with new operators for task scheduling in heterogeneous computing systems. *Engineering Applications of Artificial Intelligence*, 61(March 2016), 35–46. <https://doi.org/10.1016/j.engappai.2017.02.013>
- Borisovsky, P., Ereemeev, A., & Kallrath, J. (2020). Multi-product continuous plant scheduling: combination of decomposition, genetic algorithm, and constructive heuristic. *International Journal of Production Research*, 58(9), 2677–2695. <https://doi.org/10.1080/00207543.2019.1630764>
- Feng, Z. K., Niu, W. J., Zhou, J. Z., Cheng, C. T., Qin, H., & Jiang, Z. Q. (2017). Parallel multi-objective genetic algorithm for short-term economic environmental hydrothermal scheduling. *Energies*, 10(2). <https://doi.org/10.3390/en10020163>
- Gede, I. K., Putra, D., & Kumara, I. N. S. (2016). PENENTUAN KOMPETENSI MAHASISWA DENGAN ALGORITMA GENETIK DAN METODE FUZZY C-. 15(2).
- Hadi, R. (2017). Optimasi Fitur dalam Klasterisasi Mahasiswa Program Studi Sistem Informasi Dengan Algoritma Genetik. *Techno.Com*, 16(3), 249–255. <https://doi.org/10.33633/tc.v16i3.1392>
- Keshanchi, B., Souri, A., & Navimipour, N. J. (2017). An improved genetic algorithm for task scheduling in the cloud environments using the priority queues: Formal verification, simulation, and statistical testing. *Journal of Systems and Software*, 124, 1–21. <https://doi.org/10.1016/j.jss.2016.07.006>
- Manggolo, I., Marzuki, M. I., & Alaydrus, M. (2017). Optimalisasi Perencanaan Jaringan Akses Serat Optik Fiber To The Home Menggunakan Algoritma Genetika. *Jurnal Telekomunikasi Dan Komputer*, 2(1), 21. <https://doi.org/10.22441/incomtech.v2i1.1102>
- Mangkuto, R. A. (2017). Optimasi Perletakan Luminer Untuk Pencahayaan Ruang Industri Dengan Menggunakan Algoritma Genetik. 19(1), 1–11.
- Mauluddin, S., Ikbali, I., & Nursikuwagus, A. (2017). Optimasi Aplikasi Penjadwalan Kuliah Menggunakan Algoritma Genetik. *Jurnal Resti (Rekayasa Sistem Dan Teknologi Informasi)*, 1(1), 19–25.
- Ni Luh Gede Pivin, S., I Made, S., & Suta, D. (2016). Penerapan Algoritma Genetika Untuk Penjadwalan Mata Pelajaran.

- Journal of Applied Intelligent System*, 1(3), 220–233.
- Nitisiri, K., Gen, M., & Ohwada, H. (2019). A parallel multi-objective genetic algorithm with learning based mutation for railway scheduling. *Computers and Industrial Engineering*, 130(February), 381–394. <https://doi.org/10.1016/j.cie.2019.02.035>
- Nurhaida, I., Ramayanti, D., & Nur, I. (2019). Performance Comparison based on Open Shortest Path First (OSPF) Routing Algorithm for IP Internet Networks. *Communications on Applied Electronics*, 7(31), 12–25. <https://doi.org/10.5120/cae2019652838>
- Pezzella, F., Morganti, G., & Ciaschetti, G. (2008). A genetic algorithm for the Flexible Job-shop Scheduling Problem. *Computers and Operations Research*, 35(10), 3202–3212. <https://doi.org/10.1016/j.cor.2007.02.014>
- Rahmani Hosseinabadi, A. A., Vahidi, J., Saemi, B., Sangaiah, A. K., & Elhoseny, M. (2019). Extended Genetic Algorithm for solving open-shop scheduling problem. *Soft Computing*, 23(13), 5099–5116. <https://doi.org/10.1007/s00500-018-3177-y>
- Toscany, A. N., & Roestam, R. (2017). Pengembangan Sistem Penjadwalan Kuliah Menggunakan Algoritma Genetik (Studi Kasus : Pascasarjana Universitas Jambi). *Jurnal Manajemen Sistem Informasi*, 2(2), 379–393. <http://www.jurnalmsi.stikom-db.ac.id/index.php/jurnalmsi/article/view/120>
- Wijayaningrum, V. N., & Mahmudy, W. F. (2016). Optimization of ship's route scheduling using genetic algorithm. *Indonesian Journal of Electrical Engineering and Computer Science*, 2(1), 180–186. <https://doi.org/10.11591/ijeecs.v2.i1.pp180-186>
- Zan, X., Wu, Z., Guo, C., & Yu, Z. (2020). A Pareto-based genetic algorithm for multi-objective scheduling of automated manufacturing systems. *Advances in Mechanical Engineering*, 12(1), 1–15. <https://doi.org/10.1177/1687814019885294>
- Zhang, W., Ding, J., Wang, Y., Zhang, S., & Xiong, Z. (2019). Multi-perspective collaborative scheduling using extended genetic algorithm with interval-valued intuitionistic fuzzy entropy weight method. *Journal of Manufacturing Systems*, 53(October), 249–260. <https://doi.org/10.1016/j.jmsy.2019.10.002>
- Zhou, Z., Li, F., Zhu, H., Xie, H., Abawajy, J. H., & Chowdhury, M. U. (2020). An improved genetic algorithm using greedy strategy toward task scheduling optimization in cloud environments. *Neural Computing and Applications*, 32(6), 1531–1541. <https://doi.org/10.1007/s00521-019-04119-7>