

PENERAPAN *EVOLUTIONARY ALGORITHM* PADA PENJADWALAN PRODUKSI

(Studi Kasus di PT Brother Silver Product Indonesia)

I Gede Agus Widyadana

Dosen Fakultas Teknologi Industri, Jurusan Teknik Industri – Universitas Kristen Petra

Lala Febriana

Alumnus Fakultas Teknologi Industri, Jurusan Teknik Industri – Universitas Kristen Petra

ABSTRAK

Penelitian ini memberikan alternatif dalam menyusun suatu jadwal produksi dengan menggunakan *Evolutionary Algorithm*. Fungsi tujuan yang akan dicapai adalah meminimumkan *makespan* produksi. Metode *Shortest Processing Time (SPT)* dan *Longest Processing Time (LPT)* digunakan sebagai solusi awal. Algoritma ini kemudian diterapkan pada pabrik peralatan rumah tangga dan solusi akhir menunjukkan *Evolutionary Algorithm* memberikan *makespan* 26.74% lebih kecil dibandingkan dengan solusi awal.

Kata kunci: *Evolutionary Algorithm*, Penjadwalan.

ABSTRACT

This research gives an alternative to build production schedule using Evolutionary Algorithm. The objective function is minimizing production makespan. Shortest Processing Time (SPT) and Longest Processing Time (LPT) methods are used as initial solution. The algorithm is implemented on house ware factory and the result show the final solution has makespan 26,74% less than initial solution.

Keywords: *Evolutionary Algorithm, Scheduling.*

1. PENDAHULUAN

Penjadwalan produksi merupakan salah satu bagian yang cukup penting dalam suatu industri. Oleh sebab itu dalam beberapa dasawarsa terakhir banyak berkembang penelitian yang berkaitan dengan penjadwalan. Beberapa metode baru yang dapat diaplikasikan pada penjadwalan antara lain adalah metode *Simulated Annealing*, *Tabu Search* dan *Evolutionary Algorithm*. Metode-metode tersebut berkembang karena adanya tuntutan untuk mendapatkan jadwal produksi yang efisien dan mendekati optimal dengan jangka waktu penyusunan jadwal yang lebih singkat.

Burdett dan Kozan (2000) membangun suatu model *Evolutionary Algorithm* untuk menyelesaikan suatu masalah penjadwalan pada sejumlah stasiun kerja. Sebuah stasiun kerja terdiri dari satu atau lebih mesin yang tersusun secara paralel dan masing-masing mesin sifatnya adalah identik. Semua pekerjaan melewati urutan stasiun kerja yang sama dengan tujuan meminimumkan *makespan*.

Pada penelitian ini model yang dibangun oleh Burdett dan Kozan (2000) dicoba untuk diaplikasikan pada PT Brother Silver Products Indonesia. Perusahaan manufaktur ini bergerak pada produksi alat-alat rumah tangga yang terbuat dari logam.

2. EVOLUTIONARY ALGORITHM

Evolutionary Algorithm merupakan metode pencarian stokastik yang mencoba untuk menirukan sifat evolusi biologis dari makhluk hidup. *Evolutionary Algorithm* bekerja dengan prinsip bahwa suatu individu yang baik akan bertahan dan menghasilkan individu-individu yang semakin baik pada setiap generasi.

Elemen-elemen yang ada pada metode ini diambil dari proses alam seperti seleksi, kombinasi, mutasi, migrasi, dan perpindahan secara lokal. *Evolutionary Algorithm* bekerja lebih banyak pada sejumlah individu dalam satu populasi daripada hanya pada satu individu saja.

Proses kerja pada *Evolutionary Algorithm* meliputi seleksi yaitu proses pemilihan individu yang akan dikombinasikan untuk menghasilkan individu baru pada proses sesudahnya. Setelah dilakukan seleksi maka proses berikutnya berupa kombinasi. Proses ini berupa proses kombinasi dari individu yang telah terpilih pada proses seleksi sebelumnya. Proses ini akan menghasilkan individu baru yang berbeda dari induknya. Setelah dilakukan kombinasi pada sebagian kecil individu akan terjadi proses mutasi yaitu perubahan sifat dari suatu individu karena proses pertukaran pada individu tersebut. Selain proses-proses di atas juga dilakukan proses perhitungan nilai fungsi tujuan yang juga disebut dengan *fitness value*.

Algoritma ini berbeda dengan metode optimasi klasik yang lainnya dalam beberapa hal, antara lain :

- Metode ini merupakan metode *nondeterministic* yang akan menghasilkan penyelesaian-penyelesaian yang berbeda meskipun model awalnya tidak dirubah, dikarenakan adanya pemakaian *random sampling* dalam algoritma ini.
- Algoritma ini mempunyai populasi yang berisi calon-calon penyelesaian.
- Dalam pengaplikasiannya, *Evolutionary Algorithm* mencoba untuk menggabungkan elemen-elemen dari solusi-solusi yang telah ada untuk menciptakan solusi baru dengan mewarisi ciri-ciri yang dipunyai oleh tiap orang tua.

Pada prakteknya, *Evolutionary Algorithm* mempunyai beberapa kelemahan, antara lain:

- Terkadang suatu masalah dengan menggunakan *Evolutionary Algorithm* membutuhkan waktu yang lebih lama dibandingkan alternatif penyelesaian yang lain seperti metode *GRG (Generalized Reduced Gradient)*.
- Algoritma ini biasanya terhambat oleh dimensi dari permasalahan. Terkadang permasalahan dengan dimensi yang besar sulit mendapatkan solusi yang mendekati optimal.
- Algoritma ini tidak memiliki konsep untuk solusi optimal atau bahkan cara lain untuk menguji apakah solusi yang ditemukan itu sudah merupakan solusi yang optimal.

3. MODEL PENYELESAIAN

3.1 Individu

Individu dinyatakan dengan rangkaian urutan pekerjaan pada setiap mesin. Untuk kasus ini terdapat 10 pekerjaan yang akan dijadwalkan dan 19 buah stasiun kerja. Misalkan pada stasiun kerja satu susunan pekerjaannya adalah {2,3,10,7,9,8,1,4,6,5}.

3.2 Solusi awal

Solusi awal dibangun dengan menggunakan metode SPT (*Shortest Processing Time*) yaitu pekerjaan yang memiliki waktu pekerjaan terpendek akan dikerjakan terlebih dahulu dan LPT (*Longest Processing Time*) yaitu pekerjaan dengan proses pengerjaan terlama disusun terlebih dahulu. Kedua metode ini dipilih secara acak untuk setiap stasiun kerja.

3.3 Seleksi, Kombinasi dan Mutasi

Untuk memperkecil pengaruh probabilitas kombinasi dan mutasi maka digunakan apa yang disebut dengan *Reproduction Plan*. Dalam penelitian ini digunakan dua buah *Reproduction Plan* sebagai berikut:

1. *Reproduction Plan 1*

Memastikan bahwa setiap anggota paling sedikit sekali disilangkan dengan memilih pasangan dari populasi secara acak. Keturunan yang baru akan secara langsung dimasukkan ke dalam populasi apabila solusi tersebut tidak ada sebelumnya. Karena populasi adalah daftar yang berurutan, maka anggota yang terbaik akan berada di tempat yang pertama dan yang terburuk pada posisi yang terakhir. Anggota terakhir dipindahkan setiap kali terjadi penambahan keturunan baru. Mutasi dalam bentuk pencarian terbatas dilakukan pada tiap anggota populasi dari tiap generasi. Operator mutasi yang dapat digunakan antara lain:

1. *Inversion/reversion*, yaitu melakukan pembalikan pada suatu urutan yang dipilih dalam suatu rangkaian.
2. *Transport*, yaitu memilih suatu urutan dan menyisipkannya ke posisi yang lain dalam rangkaian tersebut.
3. *Insertion*, yaitu memilih suatu elemen, memindahkannya dan kemudian menyisipkannya ke posisi lain dalam urutan.
4. *Exchange*, yaitu memilih dua elemen dan menukar posisi mereka dalam rangkaian tersebut.

2. *Reproduction Plan 2*

Memastikan bahwa semua kemungkinan kombinasi antara setiap anggota populasi dilakukan pada setiap generasi. Apabila terdapat p anggota dalam populasi, maka akan ada $p(p-1)$ kemungkinan persilangan, sebab kombinasi antara *parent 1* dan *parent 2* akan menghasilkan *offspring* yang berbeda dengan *offspring* yang dihasilkan dari kombinasi *parent 2* dengan *parent 1*. Setiap keturunan baru yang belum ada dalam populasi secara otomatis akan dimasukkan ke dalam populasi dan anggota terburuk dihilangkan untuk menjaga agar populasi tetap pada ukuran semula. *Reproduction plan* ini tidak melakukan mutasi.

3.4 Kriteria Berhenti

Kriteria berhenti ditetapkan hingga sejumlah generasi tertentu, yang dalam penelitian ini ditetapkan sejumlah 75 generasi.

4. ANALISIS HASIL PENJADWALAN

Dalam pengerjaan penelitian ini, perhitungan penjadwalan produksi untuk meminimumkan *makespan* dilakukan dengan menggunakan program yang dibangun dari *software* Turbo Pascal 7.0. Pengolahan data dilakukan dengan 4 macam rencana reproduksi, antara lain :

1. Penggunaan *Reproduction Plan 1* hingga generasi ke-75.
Hasil perhitungan dengan *Reproduction Plan 1* menunjukkan bahwa *minimum makespan* yang dapat dicapai hingga generasi ke-75 adalah sebesar 39021.602 menit.
2. Penggunaan *Reproduction Plan 2* hingga generasi ke-75.
Hasil perhitungan dengan *Reproduction Plan 2* menunjukkan bahwa *minimum makespan* yang dapat dicapai hingga generasi ke-75 adalah sebesar 47497.453 menit.
3. Penggunaan *Reproduction Plan 1* untuk generasi ganjil dan *Reproduction Plan 2* untuk generasi genap hingga generasi ke-75.
Hasil perhitungan dengan penggunaan *Reproduction Plan 1* dan *Reproduction Plan 2* secara berselang-seling menunjukkan bahwa *minimum makespan* yang dapat dicapai hingga generasi ke-75 adalah sebesar 49654.008 menit.
4. Penggunaan *Reproduction Plan 2* untuk generasi ganjil dan *Reproduction Plan 1* untuk generasi genap hingga generasi ke-75.
Hasil perhitungan dengan *Reproduction Plan 2* dan *Reproduction Plan 1* secara berselang-seling menunjukkan bahwa *minimum makespan* yang dapat dicapai hingga generasi ke-75 adalah sebesar 44995.563 menit.

Penggunaan *Reproduction Plan 1* untuk perhitungan *makespan* menunjukkan performa yang cukup baik. Perbedaan antara nilai *makespan* pada generasi ke-1 ke generasi selanjutnya menunjukkan penurunan sedikit demi sedikit.

Reproduction Plan 2 menunjukkan performa yang baik pada awal reproduksi. Namun pada suatu saat, perhitungan dengan *Reproduction Plan 2* akan menampilkan solusi yang tidak berbeda sama sekali dengan hasil perhitungan pada generasi sebelumnya. Hal ini dapat disebabkan karena penerapan teknik *Edge Recombination* untuk *crossover*nya di mana *Reproduction Plan 2* memakai *crossover operator* ini sebagai alat utama untuk reproduksi di mana pada setiap anggota populasi dilakukan *crossover*.

Penggunaan *Reproduction Plan 1* dan *Reproduction Plan 2* secara berselang-seling dicoba untuk mengetahui apakah dengan menggabungkan kedua *Reproduction Plan* ini dapat diperoleh hasil yang lebih baik jika dibandingkan dengan penggunaan *Reproduction Plan 1* ataupun *Reproduction Plan 2* secara sendiri-sendiri. Berdasarkan nilai *makespan* terendah yang dapat dicapai, ternyata nilainya tidak berbeda sama sekali mulai dari generasi ke-6 hingga generasi ke-75. Hal ini menunjukkan bahwa solusi terbaik sudah didapatkan pada generasi ke-6.

Penggunaan *Reproduction Plan 2* dan *Reproduction Plan 1* secara berselang-seling juga dilakukan untuk mengetahui apakah dengan menggabungkan kedua *Reproduction Plan* ini dapat diperoleh hasil yang lebih baik jika dibandingkan dengan penggunaan *Reproduction Plan 1* ataupun *Reproduction Plan 2* secara sendiri-sendiri. Berdasarkan nilai *makespan* terendah yang dapat dicapai ternyata menunjukkan gejala penurunan nilai *makespan* cukup baik pada awalnya namun kemudian tidak terjadi penurunan nilai *makespan* mulai generasi ke-3 hingga generasi ke-75.

Performa untuk setiap rencana reproduksi dianalisa dengan membandingkan *makespan* terendah yang dapat dicapai hingga generasi ke-75 dengan *makespan* pada generasi awal. Penurunan yang dicapai untuk setiap penggunaan rencana reproduksi adalah sebagai berikut:

Tabel 1. Penurunan yang dicapai hingga generasi ke-75

Penggunaan rencana reproduksi	Penurunan
RP 1	26.75%
RP 2	10.84%
RP 1 (ganjil) dan RP 2 (genap)	8.56%
RP 2 (ganjil) dan RP 1 (genap)	16.47%

5. KESIMPULAN

Berdasarkan hasil simulasi dan analisa data yang telah dilakukan, maka dapat disimpulkan sebagai berikut:

1. Jadwal dengan nilai *makespan* terendah yang dapat diperoleh hingga generasi ke-75 adalah sebesar 39021.602 menit dari hasil penggunaan *Reproduction Plan 1*.
2. Penggunaan *Reproduction Plan 1* hingga generasi ke-75 menunjukkan persentase penurunan yang paling baik yaitu sebesar 26.75% dibandingkan *makespan* awal. *Reproduction Plan 1* dapat memberikan hasil yang lebih lain disebabkan adanya proses mutasi yang mampu membuat solusi lepas dari lokal optimal. Persentase penurunan ini menunjukan bahwa algoritma *Evolutionary* telah berkerja dengan baik.

DAFTAR PUSTAKA

- Blazewicz, J., et al., 1996. *Scheduling Computer and Manufacturing Processes* . Berlin : Springer.
- Burdett,R.L., and E. Kozan, 2000. “Evolutionary algorithms for flowshop sequencing with non-unique jobs”, *International Transactions In Operational Research*, VII . February, pp. 401-418.
- Company Long Name. Backgrounder: Genetic and Evolutionary Algorithms Versus Classical Optimization.[<http://www.frotnsys.com/pr11.htm>], 2000.
- Gutowitz, Howard A., 1995. *Evolutionary Algorithms* .[<http://alife.santafe.edu/alife/topics/simulators/dret/node8.html>].
- Morton, T. E., and D. W. Pentico, 1993. *Heuristic Scheduling Systems*. New York: John Wiley & Sons, Inc.