

# **PENERAPAN ALGORITMA *HYBRID CROSS ENTROPY-GENETIC ALGORITHM* DALAM PENYELESAIAN *RESOURCE-CONSTRAINED PROJECT SCHEDULING PROBLEM***

Nur Rahmawati<sup>1)</sup>, Budi Santosa<sup>2)</sup>

<sup>1)</sup> Jurusan Teknik Industri, Institut Teknologi Adhi Tama Surabaya (ITATS)  
Jl. Arif Rahman Hakim Surabaya

<sup>2)</sup> Jurusan Teknik Industri, Institut Teknologi Sepuluh Nopember  
E-mail: [rahmawatinur1987@gmail.com](mailto:rahmawatinur1987@gmail.com)

**Abstrak.** *Resource-Constrained Project Scheduling Problem (RCPSP)* adalah masalah penjadwalan proyek dengan sumber daya yang terbatas dan dengan fungsi tujuan minimasi durasi proyek. Tujuan dilakukannya penelitian ini karena problem penjadwalan proyek merupakan problem kombinatorial yang kompleks. Karena semakin banyak aktivitas dan tipe resource yang akan dijadwalkan, maka problem akan semakin sulit untuk dipecahkan. Dalam penelitian ini digunakan metode analitis untuk menyelesaikan permasalahan dengan pendekatan algoritma hybrid *Cross Entropy-Genetic Algorithm*. Berdasarkan percobaan yang telah dilakukan, maka dapat diketahui bahwa metoda CEGA sesuai apabila diterapkan pada kasus penjadwalan proyek dengan sumber daya terbatas. Kualitas solusi yang dihasilkan dengan menggunakan metoda ini dalam hal durasi terbaik, selalu menghasilkan nilai yang sama dengan nilai terbaik yang pernah ditemukan dengan metoda sebelumnya (*best known*) dengan waktu komputasi yang lebih cepat dari metoda bandingannya pada kasus kecil dan besar.

**Kata Kunci :** Hybrid CE-GA, RCPSP

## **1. Pendahuluan**

*Resource-Constrained Project Scheduling Problem (RCPSP)* merupakan problem penjadwalan proyek dengan sumber daya yang terbatas. RCPSP adalah masalah kombinatorial yang sulit untuk dipecahkan. Karena dengan bertambahnya jumlah aktivitas proyek, maka problem akan menjadi semakin kompleks. Beberapa pendekatan analitis telah dicoba seperti *Dynamic programming* [5], dan *Branch and Bound* [11] untuk mendapatkan solusi terbaik. Namun pendekatan analitis tidak dapat diterapkan apabila masalah yang diteliti sangat besar dan kompleks [7]. Oleh karena itu, beberapa metoda lain mulai diusulkan untuk memecahkan masalah tersebut.

Metoda heuristik bertujuan untuk menemukan solusi optimal dengan cara yang efisien. Sebagian besar metoda heuristik yang telah ada, menggunakan prioritas penyelesaian seperti *Shortest activity duration (SAD)*, *minimum late finish time (MILFT)*, atau *minimum total float (MITF)* untuk menentukan aktivitas yang harus dijadwalkan terlebih dahulu daripada yang lain. Namun, tidak diketahui mana prioritas yang lebih mendominasi prioritas yang lain atau yang secara konsisten memberikan performansi yang lebih baik dari yang lain [3]. Oleh karena itu, solusi yang didapatkan sering kali bersifat lokal optimal [10].

Saat ini, penyelesaian RCPSP dengan pendekatan metoda metaheuristik seperti *genetic Algorithm* [2], *ant colony optimization* [8], *particle swarm optimization* [13], dan *Harmony Search* [10] juga telah dilakukan untuk mendapatkan performansi penyelesaian problem yang lebih baik.

Metoda *Cross Entropy* sebagai salah satu metoda metaheuristik yang relatif baru telah digunakan dalam beberapa permasalahan optimasi kombinatorial, optimasi kontinu, optimasi *noisy*, dan *rare event simulation* [9]. Pada beberapa permasalahan, metoda ini dapat menghasilkan penyelesaian yang cukup optimal dengan perbandingan waktu penghitungan yang relatif lebih singkat, misalnya pada permasalahan *Support Vector Machine* [12]. Namun algoritma ini memiliki beberapa kelemahan antara lain jumlah populasi yang digunakan harus besar agar dalam populasi yang dibangkitkan tercakup solusi yang optimal. Sebagai konsekuensinya, waktu komputasi yang diperlukan semakin lama. Selain itu, algoritma ini juga memiliki kelemahan lain yaitu masih dimungkinkannya ditemukan solusi yang lebih buruk pada iterasi selanjutnya. Meskipun pada CE terdapat mekanisme *updating* parameter, namun tidak ada mekanisme pada algoritma ini yang dapat mempertahankan solusi terbaik yang telah ditemukan pada iterasi sebelumnya agar solusi tersebut tidak hilang pada iterasi selanjutnya.

## 2. Pembahasan

*Resource-Constrained Project Scheduling Problem* (RCPSP) merupakan problem penjadwalan proyek dengan sumber daya yang terbatas. RCPSP adalah masalah kombinatorial yang sulit untuk dipecahkan. Karena dengan bertambahnya jumlah aktivitas proyek, maka problem akan menjadi semakin kompleks.

### 2.1. Persamaan

Berikut ini adalah formulasi matematika dari RCPSP dengan fungsi tujuan untuk meminimasi durasi proyek [11] :

$$\text{Min}\{\text{Max } f_i \mid i = 1, 2, \dots, n\} \quad (1)$$

Subject to :

$$f_i - f_j \geq d_i; \forall j \in P_i; i = 1, 2, \dots, n \quad (2)$$

$$\sum_{A_t} r_{ik} \leq R_k; k = 1, 2, \dots, K; t = s_1, s_2, \dots, s_n \quad (3)$$

Dimana :

$N$  = jumlah aktivitas yang terdapat pada proyek

$f_i$  = waktu penyelesaian aktivitas  $i$  ( $i = 1, 2, \dots, n$ )

$d_i$  = durasi aktivitas  $i$

$P_i$  = predecessor aktivitas  $i$

$R_k$  = jumlah sumber daya tipe  $k$  yang tersedia ( $k = 1, 2, \dots, K$ )

$K$  = jumlah tipe sumber daya

$r_{ik}$  = jumlah sumber daya tipe  $k$  yang diperlukan oleh aktivitas ke-  $i$

$A_t$  = sekelompok aktivitas yang berjalan pada waktu ke- $t$

$$s_i = f_i - d_i \quad (4)$$

$s_i$  = waktu mulai aktivitas  $i$

Persamaan (1) merepresentasikan fungsi tujuan dari RCPSP. Persamaan (2) merepresentasikan *precedence constraints*. Persamaan (3) merepresentasikan *resource constraints*.

### 2.2. Algoritma CE-GA

Secara umum, mekanisme penerapan algoritma CE-GA pada RCPSP :

#### 1. Pendefinisian Nilai Input dan Output

Input yang dibutuhkan dalam penelitian ini terdiri dari: jumlah aktivitas ( $n$ ), indeks aktivitas, durasi aktivitas, kebutuhan sumber daya pada tiap aktivitas, *precedence constraints*, jumlah sumber daya yang tersedia, jumlah Populasi yang dibangkitkan ( $N$ ),  $\alpha$  (parameter penghalus),  $\rho$  (proporsi sampel elite dari keseluruhan sampel), serta toleransi pemberhentian ( $\beta$ ). Output yang dihasilkan dalam penelitian ini terdiri dari: waktu mulai dan selesai, durasi proyek, serta waktu komputasi.

#### 2. Penentuan Nilai Parameter Inisial

Nilai parameter yang harus ditentukan diawal adalah :

##### a. Jumlah Populasi yang dibangkitkan ( $N$ )

Pada penelitian ini jumlah populasi yang dibangkitkan sebesar 300 populasi.

##### b. $\alpha$ (parameter penghalus)

Nilai yang paling optimal untuk parameter penghalus 0,4 - 0,9 [4]. Nilai parameter penghalus yang digunakan pada penelitian ini adalah 0.9.

##### c. $\rho$ (proporsi sampel elite dari keseluruhan sampel)

Nilai yang paling optimal untuk parameter  $\rho$  1% - 10% [6]. Pada penelitian kali ini, digunakan  $\rho$  sebesar 5%.

##### • Toleransi pemberhentian ( $\beta$ )

Pada penelitian ini, nilai  $\beta$  yang digunakan adalah 0,0005.

##### • Parameter pindah silang (Pps)

Pada penelitian ini, parameter pindah silang yang digunakan adalah 1. Nilai tersebut didasarkan pada penelitian terdahulu [1].

##### • Parameter Mutasi (Pm)

Pada penelitian ini, parameter mutasi yang digunakan adalah setengah kali Pps. Nilai tersebut didasarkan pada penelitian terdahulu [1].

#### 3. Pembangkitan Populasi

Pembangkitan populasi inisial (iterasi=1) dilakukan dengan menggunakan teknik acak penuh (*fully randomized*), sedangkan pada pembangkitan populasi setelahnya (iterasi>1) dilakukan dengan menggunakan kaidah algoritma genetika, yakni pindah silang dan mutasi, dengan alur sebagai berikut:

- Pembobotan sampel elite

Adapun aturan pembobotan yang digunakan yakni, jika durasi yang dihasilkan oleh urutan lebih baik daripada durasi terbaik yang pernah dikunjungi pada iterasi sebelumnya, maka bobot yang diberikan adalah senilai banyaknya sampel elite, selebihnya hanya diberi bobot 1.

- Penghitungan *linear fitness ranking*

*Linear fitness ranking* (LFR) dihitung dari perbandingan nilai kesesuaian (*fitness*) semua sampel yang telah dibangkitkan pada iterasi sebelumnya. Nilai LFR adalah sebesar

$$LFR(I(N - i + 1)) = Fmax - (Fmax - Fmin) \left( \frac{i - 1}{N - 1} \right) \quad (5)$$

Dengan *i* merupakan nilai yang berkisar antara 1 sampai dengan jumlah populasi (*N*), serta *I* menunjukkan indeks aktivitas pada matriks populasi.

#### 4. Melakukan Penjadwalan Proyek dengan Skema Paralel

Berikut skemanya :

- a. Memilih sebuah aktivitas yang memiliki nilai prioritas yang lebih tinggi daripada yang lain untuk dijadwalkan tanpa melanggar *precedence constraints* dan *resource constraints*. Terlebih dahulu cek apakah masih tersedia *resource* yang cukup untuk melakukan penjadwalan aktivitas baru pada aktivitas sebelumnya yang telah dijadwalkan.
  - b. Menentukan waktu penjadwalan terbaru dimana waktu ini akan sama dengan waktu penyelesaian paling awal dari aktivitas-aktivitas yang telah dijadwalkan sebelumnya.
  - c. Apabila masih terdapat aktivitas yang belum dijadwalkan, maka proses penjadwalan kembali ke langkah 1 hingga diperoleh jadwal proyek.
5. Melakukan Perhitungan Durasi Proyek

Durasi proyek akan secara langsung diperoleh setelah dilakukan penjadwalan proyek dengan skema paralel.

6. Mengurutkan Proyek dengan Durasi yang Terkecil Hingga Terbesar

7. Pemilihan Sampel Elite

Sampel *elite* dipilih sebanyak  $[\rho N]$  dari jumlah sampel yang dibangkitkan.

8. Perbaharui parameter pindah silang dan parameter mutasi

Pembaharuan parameter dilakukan dengan menggunakan perbandingan antara rata-rata durasi pada sampel elite dengan 2 kali nilai durasi terbaik pada tiap iterasi. Nilai perbandingan tersebut didefinisikan sebagai *up*. Parameter pindah silang kemudian diperbaharui dengan menggunakan persamaan:

$$P_i = (1 - \alpha)P_{i-1} + \alpha Up \quad (6)$$

Sedangkan parameter mutasi nilainya didefinisikan sebagai  $\frac{1}{2}$  dari nilai parameter pindah silang

9. Pengecekan terhadap Kriteria Pemberhentian

10. Syarat pemberhentian pada penelitian ini adalah jika selisih antara parameter pindah silang hasil pembaharuan dengan parameter pindah silang sebelumnya lebih kecil dari  $\beta$ .

11. Penampilan Hasil

Hasil yang ditampilkan berupa: waktu mulai dan selesai tiap aktivitas, durasi proyek, dan waktu komputasi.

### 2.3. Hasil *running* Algoritma

Pengujian algoritma dilakukan dengan pembuatan kode pada software Matlab. Kemudian kode tersebut dijalankan pada kasus kecil (6 aktivitas). Spesifikasi computer yang digunakan adalah Intel Core 2 Duo, 1,66 GHz, RAM 1 GB, serta menggunakan software Matlab seri 7.10.

Tabel 1. Perbandingan hasil *running* 25 aktivitas

No. Replikasi	25 Aktivitas					
	HS		PSO		CEGA	
	Durasi	T(detik)	Durasi	T(detik)	Durasi	T(detik)
1	65	34.48	65	34.17	65	27.20
2	65	34.04	65	34.18	65	25.40
3	65	33.95	64	34.25	65	25.59
4	65	34.12	65	34.32	65	20.81
5	65	33.01	64	34.57	65	27.03
6	65	33.24	65	34.16	64	25.40
7	65	33.19	65	34.22	65	27.26
8	65	34.2	65	34.1	64	72.87
9	65	34.06	64	34.35	65	20.10
10	65	34.22	65	34.47	65	24.67
<b>Nilai terbaik</b>	<b>65</b>	<b>33.01</b>	<b>64</b>	<b>34.25</b>	<b>64</b>	<b>25.40</b>
<b>Rata-rata</b>	<b>65.2</b>	<b>33.851</b>	<b>64.7</b>	<b>34.279</b>	<b>64.8</b>	<b>29.63</b>
<b>Std. Dev</b>	<b>0.42</b>		<b>0.48</b>		<b>0.42</b>	

Tabel 2 Perbandingan hasil *running* 30 aktivitas

No. Replikasi	30 Aktivitas					
	HS		PSO		CEGA	
	Durasi	T(detik)	Durasi	T(detik)	Durasi	T(detik)
1	43	25.06	46	24.13	43	176.78
2	45	24.9	46	25.11	43	84.75
3	45	25.39	43	25.59	43	92.46
4	43	24.47	46	25.69	43	141
5	45	24.86	43	25.02	43	53.98
6	43	24.44	43	25.14	43	182.37
7	45	25	46	25.63	43	45.51
8	46	24.42	43	25.14	43	45.45
9	45	25.56	45	25.75	43	68.59
10	45	25.28	43	25	43	111.01
<b>Nilai terbaik</b>	<b>43</b>	<b>24.44</b>	<b>43</b>	<b>25</b>	<b>43</b>	<b>45.45</b>
<b>Rata-rata</b>	<b>44.5</b>	<b>24.94</b>	<b>44.27</b>	<b>25.20</b>	<b>43</b>	<b>100.19</b>
<b>Std. Dev</b>	<b>1.08</b>		<b>1.50</b>		<b>0</b>	

### 3. Simpulan

Metoda CEGA memberikan capaian durasi yang lebih baik dibandingkan dengan metoda HS pada problem kecil (25 aktivitas). Sedangkan pada problem sedang (30 aktivitas) dan besar (60 aktivitas), CEGA menunjukkan hasil durasi yang lebih stabil pada sepuluh replikasi bila dibandingkan dengan HS dan PSO. Metoda CEGA memberikan kualitas hasil capaian durasi terbaik sama seperti durasi terbaik (*best known*). Oleh karenanya, algoritma ini sesuai diterapkan pada problem RCPSP. Metoda CEGA menunjukkan performansi yang lebih bagus pada problem besar (60 aktivitas), baik dalam hal durasi maupun waktu komputasi daripada HS dan PSO.

### DaftarPustaka

- [1]. Budiman, MA 2010, 'Pendekatan *cross entropy-genetic algorithm* untuk permasalahan penjadwalan *job shop* tanpa waktu tunggu pada banyak mesin', Tugas akhir, Institut Teknologi Sepuluh Nopember, Surabaya.
- [2]. Chan, WT dan Chua, DKH 1996, 'Construction resource scheduling with genetic algorithms', Jour Constr Eng Manage ASCE, 122(2), pp. 125–32.
- [3]. Davis, EW dan Patterson, JH 1975, 'A comparison of heuristic and optimum solutions in resource-constrained project scheduling'. Manage Sci, 21, pp. 944–55.
- [4]. DeBoer, PT, Kroese, DP, Mannor, S, dan Rubinstein, RY, 2003, A tutorial on the cross-entropy method, Technion – Israel Institute of Technology, Haifa.
- [5]. Gavish, B dan Pirkul, H 1991, 'Algorithms for multi-resource generalized assignment problem', Manage Sci, 37(6), pp. 695–713.
- [6]. Kroese, DP 2009, Cross-Entropy Method, University of Queensland, Brisbane.
- [7]. Lee, JK dan Kim, YD 1996, 'Search heuristics for resource constrained project scheduling', Jour of Operat Res Soc, 47(5), pp.678–89.
- [8]. Merkle, D, Middendorf, M, dan Schmeck, H 2002, 'Ant Colony Optimization for Resource-Constrained Project Scheduling', IEEE Trans on Evolut Comp, 6, pp. 4.
- [9]. Rubinstein, RY dan Kroese, DP 2004, The Cross-entropy method: A unified approach to combinatorial optimization, monte-carlo simulation, and machine learning, Springer Science+Business Media Inc, New York.
- [10]. Setiawan, A 2010, 'Penerapan Algoritma Harmony Search Dalam Penyelesaian Resource-Constrained Project Scheduling Problem', Tugas akhir, Institut Teknologi Sepuluh Nopember, Surabaya.
- [11]. Stinson JP 1976, 'A branch and bound algorithm for general class resource-constrained schedulling problem', PhD thesis, University of North Carolina at Chapel Hill, Chapel Hill.
- [12]. Widyarini, T 2009, 'Aplikasi Metode Cross Entropy untuk Support Vector Machines', Tugas akhir, Institut Teknologi Sepuluh Nopember, Surabaya.
- [13]. Zhang, H, Li, H, dan Tam, CM 2006, 'Particle swarm optimization for resource-constrained project scheduling'. Intern Jour of Proj Manage, 24, pp. 83–92.