

Penyelesaian Permasalahan Penjadwalan Aktivitas Proyek dengan Batasan Sumber Daya Menggunakan Metode Cross Entropy

Problem Solving on The Resource Constrains Project Scheduling Problem (RCPSP) Using Cross Entropy Method

Maria Krisnawati

maria.krisnawati@unsoed.ac.id

Program Studi Teknik Industri, Universitas Jenderal Soedirman, Purwokerto

Abstrak— Penjadwalan aktivitas proyek merupakan hal yang penting untuk direncanakan sebelum melakukan aktivitas proyek. Metode penjadwalan proyek standar menjadwalkan aktivitas berdasarkan *precedence constraints* dan asumsi bahwa kapasitas sumberdaya tidak terbatas. Namun pada kenyataannya aktivitas proyek mempunyai batasan sumber daya. Tujuan dari penelitian ini adalah meminimalkan total durasi proyek dengan *precedence constraints* dan *resource constraints* untuk seluruh aktivitas proyek. Banyak metode optimasi yang digunakan untuk meningkatkan kualitas penjadwalan dan mempercepat waktu penghitungan. Pada penelitian ini mengimplementasikan metode *Cross Entropy* (CE) dalam permasalahan penjadwalan aktivitas proyek dengan batasan sumber daya, serta membandingkan kelebihan dan kekurangan antara metode CE dengan metode *Differential Evolution* (DE). CE terdiri atas empat langkah utama yaitu, menentukan parameter awal, membangkitkan sampel secara random berdasarkan parameter, menghitung performansi berdasarkan fungsi, memilih sampel elit dan memperbaharui parameter sebelumnya untuk mendapatkan sampel yang lebih baik pada iterasi selanjutnya. Untuk mempercepat waktu komputasi, pada penelitian ini menurunkan jumlah sampel untuk setiap iterasi dalam optimasi total dengan algoritma *Cross Entropy*. Eksperimen Numerik dengan beberapa set data dari *Project Scheduling Problem Library* (PSPLIB) memperlihatkan bahwa CE dapat memberikan perhitungan total durasi proyek optimal yang sama dengan DE, sedangkan waktu perhitungannya lebih cepat daripada DE.

Kata kunci— Penjadwalan aktivitas proyek, *Cross Entropy*, optimasi

Abstract— Project scheduling is essential to be planned before activities. Standard methods of project scheduling based on *precedence constraints* scheduling events and the assumption that resources are not unlimited capacity. But in fact the project activity has resource limits. The main objective is to minimize the total duration of the project with *precedence constraints* and *resource constraints* for all project activities. Many optimization methods are used to improve the quality of scheduling and speed up the calculation time. This paper proposed the use of *Cross Entropy* (CE) method to solve resource constraints project scheduling problem, as well as comparing the advantages and disadvantages between the CE methods with *Differential Evolution* (DE) method. The purpose CE method consist of four critical steps including generating a sample of random solution, calculating the performance according to the specific fitness function, selecting elite sample and finally updating the previous parameters to get a better sample in the next iteration. To speed up the computation time, in this study decrease the number of samples for each iteration in *Cross Entropy* algorithm. Numerical experiments with several data sets from the *Project Scheduling library* (PSPLIB) showed that CE can provide the optimal total duration of the project same as the DE with calculation time is faster than DE.

Keyword— Project Scheduling, *Cross Entropy*, Optimization

PENDAHULUAN

Pada manajemen proyek, sebelum proyek dikerjakan perlu adanya tahap – tahap pengelolaan proyek yang meliputi tahap perencanaan, tahap penjadwalan, dan tahap pengkoordinasian. Dari ketiga tahap ini tahap perencanaan dan penjadwalan adalah tahap yang paling menentukan berhasil atau tidaknya suatu proyek.

(Arifudin, dkk, 2014). Pada tahap penjadwalan proyek, umumnya digunakan pendekatan *critical path method* (CPM). Penerapan CPM membutuhkan data aktivitas (pekerjaan), durasi dan hubungan ketergantungan antar pekerjaan. Dengan CPM akan diketahui berapa umur proyek (*makespan*) terpendek. Pemakaian CPM menggunakan asumsi tidak ada keterbatasan sumber daya. Sehingga pemakaian CPM akan tidak lagi layak

jika ada konstrain tambahan berupa ketersediaan sumberdaya. Jika terjadi keterbatasan sumberdaya untuk jadwal yang sudah diperoleh dengan CPM, perlu dilakukan lagi pengaturan jadwal menyesuaikan ketersediaan sumberdaya yang dimiliki. Atau perlu ditambahkan sumberdaya agar jadwal yang didapatkan bisa berlaku. (Santosa dan Willy, 2011) Penentuan jadwal proyek dengan keterbatasan sumberdaya ini yang sering dikenal dengan *Resource Constrains Project Scheduling Problem (RCPSP)* akan menjadi cukup sulit jika aktivitas proyek membesar. Oleh karena itu, perlu pendekatan metaheuristik untuk penyelesaian permasalahan tersebut.

Resource Constrains Project Scheduling Problem (RCPSP) adalah penjadwalan aktivitas proyek dengan *precedence constrain* dan *resource constrain*. *Precedence constraints* adalah suatu kendala dimana aktivitas pendahulu harus sudah selesai dijadwalkan sebelum aktivitas yang lain dijadwalkan. Sedangkan *Resource constraints* adalah suatu kendala dimana sumber daya yang diperlukan oleh setiap aktivitas pada setiap unit waktu tidak boleh melebihi kapasitas sumber daya yang tersedia. Tujuan dari penyelesaian RCPSP adalah meminimalkan waktu durasi proyek. Berbagai metode metaheuristik telah digunakan untuk menyelesaikan RCPSP diantaranya *Genetic Algorithm /GA* (Putra dan Agung, 2010; Suliantoro dkk, 2008; Arifudin, 2011; Sakalauskas dan Felinskas, 2006; serta Buliali dkk, 2008), Metode *Linear programming* (Brucker dan Knust, 2000; Damay, 2007; Herroelen dkk, 1998), *Hybrid scatter search* (Ranjbar dkk, 2009), *Tabu Search Approach* (Thomas dan Salhi, 1998), *ant colony optimization* (Xiao dkk., 2013; serta Liang dkk., 2004), *differential evolution* (Damak, 2009), metode *cross entropy* (Jedrzejowicz dan Skakovski, 2013).

Dari penelitian sebelumnya, belum banyak penelitian untuk RCPSP yang menggunakan metode *cross entropy*. Dalam metode *cross entropy* penentuan sampel menjadi kunci sukses dalam efisiensi perhitungan. Pengembangan metode *cross entropy* yang digunakan oleh Jedrzejowicz dan Skakovski (2013) mempelajari bagaimana topologi interkoneksi dan heterogenitas modul konstituen mempengaruhi efisiensi. Pada penelitian ini akan digunakan pengurangan sample pada metode *cross entropy* untuk penyelesaian RCPSP. Sebelumnya metode *cross entropy* dengan pengurangan jumlah sampel di setiap iterasi telah dilakukan oleh Krisnawati, dkk. (2011) pada permasalahan kombinatorial yang cukup sulit untuk menyelesaikan *crew rostering* dan *Flow Shop Scheduling*. Hasil dari perhitungan numerik menunjukkan metode *cross entropy* dengan pengurangan sample menunjukkan performansi yang cukup baik untuk permasalahan kombinatorial yang besar (*NP-Hard Problem*) jika dibandingkan dengan metode CE tanpa penurunan jumlah sampel, *Differential*

Evolution dan PSO baik dalam hasil penjadwalan yang diperoleh maupun dari segi waktu perhitungan.

Metode *Differential evolution (DE)* termasuk metoda pencarian stokastik dan berdasarkan populasi (*population based search*). Dan sejak diusulkan 1995, DE mendapatkan reputasi yang bagus sebagai *global optimizer* yang efektif. (Santosa dan Willy, 2011). Oleh karena itu pada penelitian ini akan dibandingkan penggunaan *cross entropy* dengan pengurangan jumlah sampel untuk setiap iterasi dengan *differential evolution*. Metode DE dan CE dengan pengurangan jumlah sampel akan dibandingkan dari segi waktu durasi proyek yang dihasilkan dan waktu perhitungan.

DEFINISI PERMASALAHAN

Keseluruhan aktivitas proyek dinamakan dengan *activity on node (AON) network* dimana setiap node merepresentasikan aktivitas, durasi pekerjaan dan aktivitas sebelum dan setelahnya. Setiap aktivitas yang dijadwalkan harus memenuhi hubungan antar aktivitas dan ketersediaan sumber daya untuk setiap periode waktu. Model matematis untuk permasalahan penjadwalan proyek dengan sumber daya terbatas (Liang dkk., 2004), dapat dirumuskan sebagai berikut :

$$\text{Min} \left[\text{Max}_{j \in J} (t_a \times x_{jt_a}) \right] \quad (1)$$

Dengan batasan sebagai berikut.

$$\sum_{t_a=EF_j}^{LF_j} x_{jt_a} = 1 \quad j = 1, \dots, J \quad (2)$$

$$\sum_{t_a=EF_h}^{LF_h} t_a \times x_{ht_a} \leq \sum_{t_a=EF_j}^{LF_j} (t_a - d_j) \times x_{jt_a} \quad h \in P_j, J = 2, \dots, J \quad (3)$$

$$\sum_{j=1}^J k_{jr} \sum_{q=\max\{t_a, EF_j\}}^{\min\{t_a+d_j-1, LF_j\}} x_{jq} \leq K_r \quad r \in R, t_a = 1, \dots, T^* \quad (4)$$

$$x_{jt_a} \in \{0,1\} \quad j = 1, \dots, J, t_a = EF_j, \dots, LF_j \quad (5)$$

Keterangan :

J	aktivitas
j	indeks untuk aktivitas, j = 1, 2, ..., J
R	sumber daya
r	indeks untuk sumber daya r ∈ R
ES _j	waktu paling awal dapat dimulainya aktivitas j
LS _j	waktu paling akhir dapat dimulainya aktivitas ke j
EF _j	waktu paling awal untuk aktivitas j selesai dikerjakan
LS _j	waktu paling akhir untuk aktivitas j selesai dikerjakan
T*	batasan durasi proyek
K _r	jumlah unit sumberdaya r, r ∈ R, yang ada pada saat t _a , t _a = 1, ..., T*
P _j	sekumpulan aktivitas yang harus dilakukan sebelum memulai aktivitas j
d _j	durasi aktivitas j
k _{jr}	jumlah unit sumber daya r, r ∈ R, yang digunakan di aktivitas j

$$x_{jt_a} = \begin{cases} 1 & \text{jika aktivitas } j \text{ diselesaikan pada akhir periode } t_a \\ 0 & \text{sebaliknya} \end{cases}$$

Persamaan (1) meminimasi durasi proyek sebagai tujuan utama. Persamaan (2) memastikan bahwa hanya satu kali waktu penugasan untuk setiap aktivitas, dan hubungan antar aktivitas dipastikan oleh persamaan (3). Persamaan (4) mendeskripsikan penggunaan tipe sumber daya untuk aktivitas yang dijadwalkan tiap unit waktu tidak dapat melebihi kapasitas sumber daya yang ada. Kemudian persamaan (5) mendefinisikan variable biner. Dengan kata lain, batasan durasi proyek (T^*) merupakan jumlah dari durasi aktivitas maksimum.

Beberapa asumsi yang digunakan dalam RCSP antara lain sebagai berikut.

- Durasi aktivitas diketahui dan tetap.
- Tidak ada tindakan pencegahan aktivitas.
- Adanya hubungan antar aktivitas.
- Jumlah dan tipe sumber daya yang dibutuhkan untuk setiap aktivitas diketahui dan tetap.
- Durasi aktivitas dan sumber daya yang ada merupakan bilangan integer dan terpisah.
- Aktivitas hanya dapat dilakukan jika mode telah ditentukan, misalnya mode tunggal.

ALGORITMA CROSS ENTROPY

Metode *cross entropy* diilhami oleh sebuah konsep pada teori informasi modern, yakni konsep jarak Kullback-Leibler atau yang dikenal juga dengan nama yang sama : konsep jarak *cross entropy* (Rubinstein dkk, 2004). Metode CE awalnya digunakan sebagai alat untuk mengestimasi probabilitas dari kejadian langka (*rare event*) dengan penerapan algoritma adaptive untuk pada peristiwa stokastik yang kompleks, dengan cara meminimasi variansi (*variance minimization*). Pada perkembangan selanjutnya, ditemukan bahwa modifikasi sederhana terhadap metoda CE dapat digunakan tidak hanya untuk mengestimasi probabilitas kejadian langka, tetapi juga untuk menyelesaikan permasalahan optimasi kombinatorial yang kompleks dengan cara meminimasi *cross entropy*. (Santosa dan Willy, 2011). Masalah optimasi kombinatorial ini biasanya memerlukan waktu penghitungan yang lama jika diselesaikan dengan teknik optimasi eksak seperti *integer programming*, terutama jika ukuran problem membesar. Masalah *travelling salesman problem*, *vehicle routing problem* atau *bin packing* adalah contoh-contoh optimasi kombinatorial. Metode random yang telah dikenal untuk menyelesaikan masalah optimasi kombinatorial diantaranya ialah *simulated annealing*, *genetic algorithm*, *tabu search* dan beberapa lagi. Metoda CE melibatkan prosedur iterasi, dimana tiap iterasi dapat dipecah menjadi dua fase:

1. Melakukan pembangkitan sampel random (x) dengan menggunakan mekanisme atau distribusi tertentu.
2. Memperbaharui parameter (v) dari mekanisme random berdasarkan data sampel elite untuk menghasilkan sampel yang lebih baik pada iterasi berikutnya. (Santosa dan Willy, 2011) Pada penelitian ini juga ditambahkan adanya *decreasing sampel* untuk setiap iterasi (Krisnawati, dkk, 2011).

Algoritma *cross entropy* dengan penurunan jumlah sampel untuk permasalahan RCSP adalah sebagai berikut:

(i) Set initial parameter :

- Set percentile of elite sample, ρ ;
- number of population, $npop$;
- generation number, t ;
- maximum generation, t_{max} .

(ii) Pembangkitan sampel

a. Untuk iterasi awal ($t = 0$)

Untuk iterasi awal dibangkitkan sejumlah solusi awal yang berupa jadwal aktivitas j terhadap waktu t_a sejumlah $npop$ berdasarkan parameter \hat{p}_0 . Setiap elemen dari matriks parameter \hat{p}_0 yang berukuran $m \times n$ mempunyai probabilitas sukses masing – masing sebesar $1/2$, dengan m adalah jumlah aktivitas dan n adalah waktu t_a . Pada iterasi awal semua aktivitas mempunyai probabilitas sukses yang sama untuk ditempatkan pada suatu urutan tertentu. Solusi aktivitas j yang berukuran $n \times m$ dibangkitkan sebanyak $npop$ (untuk $npop = 1000, 2000, 3000$ dan 4000).

b. Untuk iterasi selanjutnya ($t \neq 0$)

Pembangkitan sampel untuk iterasi selanjutnya bukan lagi berdasarkan \hat{p}_0 melainkan berdasarkan probabilitas sukses \hat{p}_t , dimana \hat{p}_t adalah parameter yang telah diupdate pada setiap iterasi ke – t menggunakan data sample elite. Solusi jadwal aktivitas berukuran $n \times m$ dibangkitkan sebanyak $npop$ (untuk $npop = 1000, 2000, 3000$ dan 4000).

(iii) Pembaharuan Parameter $p0$

Hitung nilai *fitness* ($\frac{1}{(1+makespan)}$) dan urutkan dari yang terkecil hingga yang terbesar. Selanjutnya tentukan sampel elite. Parameter $p0$ diperbaharui berdasarkan data sampel elite untuk mendapatkan sampel yang lebih baik dari iterasi sebelumnya dengan :

$$\hat{p}_t = \alpha \tilde{\omega}_t + (1 - \alpha) \hat{p}_{t-1} \quad (6)$$

dimana,

$\tilde{\omega}_t$ adalah rata – rata sampel elite

α adalah parameter *smoothing* yang bernilai ($0 < \alpha < 1$), pada penelitian ini menggunakan α (α) = 0.9.

(iv) Pengecekan terhadap Syarat Pemberhentian

Syarat pemberhentian pada penelitian ini adalah *maxit* (*maxit* = 100 iterasi). Jika syarat pemberhentian ini terpenuhi, maka hentikan iterasi dan jika tidak kembali ke langkah (ii).

(v) Decreasing sample

Pada tahap ini dilakukan penurunan jumlah sampel untuk iterasi berikutnya. Jumlah populasi sampel untuk iterasi berikutnya sebanyak :

$$npop(it + 1) = decreasing\ sample \times npop(it) \quad (7)$$

Selanjutnya, ulangi kembali iterasi mulai langkah (ii).

DIFFERENTIAL EVOLUTION (DE)

Algoritma DE Mutasi DE pada penelitian ini umumnya menggunakan cara mutasi *random swap*. $r0$ adalah bilangan random antara 0 dan 1 yang berdimensi $m \times n$ untuk setiap populasi np . Solusi V terdiri dari pasangan $X_{np,g}^u$. vnp,ro,g adalah elemen solusi V kolom $r0$ generasi g , jika $Wnp,g-1$ adalah populasi terbaik generasi $g-1$ dan $w_{np,r0,g-1}$ adalah elemen kolom $r0$ dari $Wnp,g-1$.

a. Crossover

Crossover adalah menyilangkan atau menukarkan solusi induk Xnp,g dengan solusi mutan Vnp,g untuk membentuk solusi baru Unp,g . Caranya adalah dengan menentukan probabilitas *threshold* ($0 < cr < 1$) mutan menjadi solusi baru. Kemudian dilakukan randomisasi sebanyak n_pop . Jika bilangan random terjadi antara $0 < randnp(0,1) < cr$, maka mutan menjadi solusi baru, yang lain solusi induk akan menjadi solusi baru.

b. Selection

Proses ini dilakukan dengan cara membandingkan solusi induk dengan solusi baru hasil *crossover*. Populasi induk yang memberikan performansi lebih baik dari pada populasi solusi hasil *crossover* akan dipertahankan, sebaliknya populasi hasil *crossover* akan menggantikan populasi induk pada iterasi selanjutnya. Solusi generasi berikutnya yaitu $X_{np,g+1}^u$ diperoleh dengan formulasi berikut :

$$X_{np,g+1}^u \begin{cases} U_{np}^u & \text{jika } f(U_{np,g}^u) < f(X_{np,g}^u) \\ X_{np,g}^u, & \text{sebaliknya} \end{cases} \quad (8)$$

$f(U_{np,g}^u)$ adalah nilai fungsi *fitness* dari $U_{np,g}^u$,

sedangkan $f(X_{np,g}^u)$ adalah nilai fungsi *fitness* dari $X_{np,g}^u$.

c. Stopping Criterion

Proses akan dihentikan jika iterasi sudah mencapai iterasi maksimum (*gmax*).

PERCOBAAN DAN ANALISA NUMERIK

Data yang digunakan untuk pengujian algoritma adalah data set dari *Project Scheduling Problem library* (*PSPLIB, 1996*) dengan jumlah aktivitas 30, 60, 90 dan 120. Untuk jumlah aktivitas proyek 30 dan 60 dilakukan pengujian masing – masing 20 kali pengulangan, sedangkan untuk jumlah aktivitas proyek 90 dan 120 dilakukan masing – masing 10 kali pengulangan. Hasil perhitungan Durasi Proyek dan waktu komputasi terbaik untuk kedua algoritma yang digunakan untuk setiap data set terlihat pada Tabel 1.

Dari hasil perbandingan perhitungan durasi proyek dengan kedua metode, memperlihatkan bahwa penggunaan algoritma *cross entropy* menunjukkan performansi yang relatif cukup baik untuk permasalahan RCPSP dengan jumlah aktivitas yang cukup besar jika dibandingkan dengan metode *differential evolution*. Sedangkan untuk jumlah aktivitas 30 dan 60 pencarian durasi proyek dengan metode *cross entropy* menunjukkan performansi yang sama dengan penyelesaian menggunakan *differential evolution*. Waktu komputasi perolehan jadwal dengan durasi proyek minimum dengan metode *cross entropy* juga jauh lebih cepat jika dibandingkan dengan metode *Differential Evolution*. Hal ini dapat dilihat pada Tabel 2.

Tabel 1. Perbandingan Durasi Proyek Metode *Cross Entropy* dan *Differential Evolution*

Data Set	Jumlah Aktivitas	Cross Entropy (CE)	Differential Evolution (DE)
J3013	30	32	32
J6029	60	57	57
J12016	120	77	78

Tabel 2. Perbandingan Waktu Komputasi Penyelesaian Permasalahan Penjadwalan Aktivitas Proyek dengan Metode *Cross Entropy* dan *Differential Evolution* (detik)

Data Set	Cross Entropy (CE)	Differential Evolution (DE)
30	1.5719	8.1057
60	5.7177	25.3899
120	30.8688	100.0760

KESIMPULAN DAN SARAN

Metode CE dan DE telah dikembangkan untuk menyelesaikan masalah penjadwalan aktivitas proyek dengan batasan sumber daya. Metode CE telah menunjukkan hasil yang relatif sama dengan metode DE

dalam hal durasi proyek. Hal ini sangat mampu untuk mendapatkan solusi yang mendekati optimal dengan waktu komputasi yang relatif cepat untuk metode CE. Penurunan jumlah sampel meningkatkan efektivitas algoritma dengan penurunan sampel sehingga mempercepat waktu komputasi.

Untuk penelitian mendatang, dapat digunakan algoritma yang kompleks untuk penyelesaian model RSPCP dengan jumlah aktivitas yang besar, misalnya dengan metode CE hibrida dengan metode heuristik lain untuk mendapatkan solusi yang lebih baik dan waktu komputasi yang lebih cepat.

DAFTAR PUSTAKA

- Arifudin, R., Widyawati, K., Mashuri M., 2014, Analisis Algoritma Branch and Bound untuk menyelesaikan Masalah Penjadwalan Proyek Pembangunan Mega Tower, Unnes Journal of Mathematics, Vol 3, No. 1.
- Brucker, P. and Knust S., 2000, A linear programming and constraint propagation-based lower bound for the RCPSP, European Journal of Operational Research, Volume 127, Number 2, pp 355-362.
- Buliali, Lianto, J., Herumurti, D., dan Wiriapradja, G., 2008, Penjadwalan Matakuliah dengan Menggunakan Algoritma Genetika dan Metode *Constrain Satisfaction*, Jurnal Ilmiah Teknologi Informasi, Volume 7, Number 1 pp 29-38.
- Damak, N., 2009, Differential evolution for solving multi-mode resource-constrained project scheduling problems. Computers & Operations Research, Volume 36, Number 9, pp. 2653-2659.
- Damay, J., Quilliot, A., and Sanlaville E., 2007, Linear programming based algorithms for preemptive and non-preemptive RCPSP, European Journal of Operational Research, Volume 182, Number 3, pp. 1012-1022.
- Herroelen, W., Reyck, B. D., dan Demeulemeester E., 1998, Resource-constrained project scheduling : a survey of recent developments, Computers & Operations Research, Volume 25, Number 4, pp. 279-302.
- Jedrzejowicz, P., dan Skakovski, A., 2013, Population learning with differential evolution for the discrete-continuous scheduling with continuous resource discretisation, Cybernetics (CYBCONF), IEEE International Conference on IEEE.
- Krisnawati, M., Santosa, B., dan Rusdiansyah, A., 2011, Comparison Of Cross Entropy And Differential Evolution To Solve Crew Rostering Problem, International Engineering and Service Science (IESS) Conference, Surakarta, Indonesia.
- Liang, Y.C., Chen, A. H. L., Chen, Wen-Ching Kao, dan Chyu, C.-C., 2004, An ant colony approach to resource-constrained project scheduling problems, Proceedings of the 5 th Asia Pacific Industrial Engineering and Management Systems Conference 2004.
- Putera, A., dan Agung, I.G., 2010, Proses Model Penjadwalan Proyek dengan Algoritma Genetika, Jurnal Ilmiah Teknik Sipil, Vol. 14, No. 1, Januari 2010.
- Ranjbar, M., Reyck, B. D., and Kianfar, F., 2009, A hybrid scatter search for the discrete time/resource trade-off problem in project scheduling, European Journal of Operational Research, Volume 193, Number 1, pp. 35-48.
- Rubinstein, Reuven, Y., dan Kroese, Dirk P., 2004, The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation, and Machine Learning, New York : Springer Science+Business Media, Inc.
- Sakalauskas, L. and Felinskas, G. , 2006, Optimization of Resource Constrained Project Schedules By Genetic Algorithm Based on Priority List, Information Technology and Control, Volume 35, Number 4, pp 412 – 418.
- Santosa, B., dan Willy, P., 2011, Metoda Metaheuristik Konsep dan Implementasi, Surabaya : Guna Widya.
- Suliantoro, H., Fanani, R., dan Handayani, A., 2008, Optimasi Penjadwalan Proyek Dengan Keterbatasan Sumber Daya Manusia Menggunakan Algoritma Genetika (Studi Kasus Proyek M000229 PT PAL Indonesia), Tugas Akhir Teknik Industri UNDIP, Tidak dipublikasikan.
- Thomas, P. R., and Salhi, S., 1998, A tabu search approach for the resource constrained project scheduling problem, Journal of Heuristics, Volume 4, Number 2, pp. 123-139.
- Xiao, J., Ao, X.T., and Tang, Y., 2013, Solving software project scheduling problems with ant colony optimization, Computers & Operations Research, Volume 40, Number 1, pp. 33-46.
- PSPLIB – A project Scheduling Library, <http://www.bwl.uni-kiell.de/Prod/psplib/>, 1996.