SOLVING ASSEMBLY LINE BALANCING PROBLEM USING GENETIC ALGORITHM TECHNIQUE WITH PARTITIONED CHROMOSOME

Nora Azmi[1] , Iman Yahya Azzubaidi[2], Sumiharni Batubara[3].

[1,2,3]Industrial Engineering Department, Trisakti University, Indonesia
E-mail address : noraazmi@yahoo.com

ABSTRACT

In this research, the settlement of a Simple Assembly Line Balancing Type E (SALBP-E) was done by Genetic Algorithm (GA) technique. Although there are many researches about SALBP-E was solved by GA, but formation of the chromosomes in the assembly line which presented by a large network diagram is not a simply problem and still have limited discussion. In this case, chromosome was partitioned based on relationship order of the network diagram. This makes most of child chromosomes (offsprings) resulted from the crossover and mutation process were a feasible solution, so that optimal solution can be accomplished more quickly. Genetic operation was performed by using Borland Delphi program. The results showed an increasing of the line balancing efficiency and the production targets was achieved.

Key words : *SALBP-E, Genetic algorithm with partitioned chromosome*

1. INTRODUCTION

Assembly lines is one of the important facilities at mass-manufacturing industries. On the assembly line, the components are assembled into a finished products. Performance of assembly lines determine a company's ability to achieve production targets. However, often the manufacturers are not able to achieve the production target due to several constraints on the assembly line.

Company X manufactures various types of refrigerator consisting of 1 door refrigerator and 2-door refrigerator. The company currently faces problems of not achieving production targets due to the imbalance of the workload on the cooling unit assembly lines, where there is not the same process time between work stations. It causes bottlenecks at several work stations while several other stations are idle.

Assembly lines problem can be classified into two major groups, that are the Simple Assembly Line Balancing Problem (SALBP) and the Generalized Assembly Line Balancing Problem (GALBP).

SALBP limited by several assumptions, such as : 1) Mass production of one homogenous product; 2) All tasks are processed in a predetermined mode (no processing alternatives exist) 3) paced line with fixed cycle time c; 4) deterministic (and integral) operation times t_j; 5) no assignment restrictions besides the precedence constraints; 6) serial line layout with m one-sided sta tions; 7) all stations are equally equipped with respect to machines and workers; and 8) maximize the line efficiency. Meanwhile GALBP is a problem type which generalize or remove some assumptions of SALBP so assembly line problem solving is closer to real condition (Becker and Scholl, 2006; Kriengkorakot and Pianthong, 2007).

SALBP can be classified by its objective function where the problem versions include the SALBP -1, SALBP-2, SALBP-F and SALBP-E [1], [2]. The objective of the SALBP-1 problem is to minimize the number of workstations for a given cycle time, whereas SALBP-2 problem minimizes the cycle time given apredetermined number of workstations. Unlike the previous two versions, SALBP-F determines whether or not a feasible assembly configuration exists for a given combination of cycle time and number of workstations. Lastly, the SALBP-E attempts to maximize the line efficiency by minimizing the number of workstations and cycle time simultaneously (Boysen, Fliedner

and Scholl, 2007; Chong, Omar, and Abu Bakar, 2008)

The solution for SALBP problem can be obtained through the exact procedures such as Branch and Bound method and Linear Programming; and heuristic methods such as Positional weight method, Kilbridge and Wester. In the last 20 years widely used metaheuristics approach like a Tabu Search, Simulated Annealing, Genetic Algorithm and Ant Colony. Among these methods, GA is the most widely used to solve SALBP problem for past 10 years. Researchers like used GA in ASP problem because it can generate optimum or near optimum solution faster than exact algorithms. In GA, the number of considered solution is reduced compared with exact algorithms (Rashid, Hutabarat and Tiwari, 2011).

Although many SALBP cases were resolved by GA approach, but there are some obstacles in applying GA in the case of ALBP. One of them was the formation of chromosomes representing precedence network of the assembly line problem. (Scholl and Becker 2006). The improper chromosome can resulted an infeasible offsprings that violated precedence constraints. In this study, GA is applied to SALBP-E problem through the formation a partitioned chromosomes to reduce infeasible solution and .accelerate the achievement an optimal or near optimal solution.

2. THEORETICAL BACKGROUND

2.1. Assembly Line Balancing Problem (ALBP)

Any type of ALBP consists in finding a feasible line balance, i.e., an assignment of each task to exactly one station such that the precedence constraints and possibly further restrictions are fulfilled. The set S_k of tasks assigned to a station k (=1, . . .,m) constitutes its station load, the cumulated task time $t(S_k) = \sum_{j \in S_k} t_j$ is called station time. When a fixed common cycle time c is given, a line balance is feasible only if the station time of neither station exceeds c (Scholl & Becker, 2006).

According Chase et al (1998), the steps in balancing an assembly line are :

1) Specify the sequential relationships among tasks using a precedence diagram.
2) Determine the required cucle time (C), using the formula :
$$C = \frac{Production\ time\ per\ day}{Required\ output\ per\ day\ (in\ units)} \quad (1)$$
3) Determine the theoretical minimum number of workstation (N_t) required to satisfy the cycle time constraint using the formula :
$$N_t = \frac{Sum\ of\ task\ times\ (T)}{Cycle\ time\ (C)} \quad (2)$$
4) Select a procedure (method) by which tasks are to be assigned to workstations.
5) Assign tasks, one at a time, to the first workstation until the sum of the task times is equal to the cycle time, or no other tasks are feasible because of time or sequence restrictions. Repeat the process for workstation 2, workstation 3, and so on, until all tasks are assigned.
6) Check whether the goals of assembly line balancing has been reached. If the goals are unsatisfactory, rebalance using a different rule (procedure).

The goals of the ALBP problems are to minimize the number of workstations (m), minimize the workload variance (w_v), to minimize the idle time (T_{id}), maximize the line efficiency (E) and minimize the smoothness index (SX) (Suwannarongsri and Puangdownreong, 2008; Scholl and Becker, 2006). Given number of tasks (n), cycle time (c), total processing time (W) and the processing time of the i^{th} workstation (T_i), the goals of assembly line problem can be determined as follows (Suwannarongsri and Puangdownreong, 2008) :

$$m = \frac{W}{c} \quad (3)$$
$$T_{id} = \sum_{i=1}^{m}(c - T_i) \quad (4)$$
$$w_v = \sum_{i=1}^{m}[T_i - (W/m)]^2/m \quad (5)$$
$$E = \sum_{i=1}^{m} T_i/(mc) \quad (6)$$
$$SX = \sqrt{\sum_{i=1}^{m}(c - T_i)^2} \quad (7)$$

SALBP-E may be solved by some exact methods and heuristic methods. Such a method has to find a feasible combination (m, c) of the number m of stations and the cycle time c such that the line efficiency is maximized or, equivalently, the required line

capacity T = m.c is minimized (Scholl and Becker, 2006).

2.2. Genetic Algorithm for Solving SALBP

GA is widely used to solve the assembly line balancing problem for the past 20 years

The basic GA is as follows (Sivanandam & Depa, 2008) :

- Generate initial populations of chromosomes.
- Evaluate the fitness of each chromosome in the population.
- Create a new population by repeating following steps until the new population is complete :
  - Select two parent chromosomes from a population according to their fitness.
  - With a crossover probability, cross over the parents to form new offspring (children). If no crossover was performed, offspring is the exact copy of parents.
  - With mutation probability, mutate new offspring at each locus (position of chromosome)
  - Place new offspring in the new population.
- Use new generated population for a further sum of the algorithm.
- If the end condition is satisfied, stop, and return the best solution in current population.
- Go to step 2 for fitness evaluation.

Although there are numerous papers that used this algorithm, GA in an original and basic form is unsuitable to be directly used to solve and optimise ASP and ALB problems. There are three issues have to be resolved (Scholl and Becker, 2006; Rashid et al, 2011) :

- How to set the proper encoding for chromosomes that will represent tasks contained in the precedence network.
- Define the fitness function that give a strong distinction between the solution's fitness. To illustrate, the objective function to minimizing workstations at SALBP-1 tends to give the same solutions so that the convergence of the genetic operations quickly reached.
- Running the genetic operators such as crossover and mutation, so the number of infeasible solutions and breaking precedence can be reduced.

Tseng (2006) proposed three aspects to improve traditional GA :

- Setting feasible initial population before the evolution process starts, therefore, efficiency and quality of the solutions can be improved.
- In traditional GAs, the crossover procedure is often randomly treated. Without consideration of limitations of the assembly problem, the procedure will generate a large number of infeasible solutions during the evolutionary stage. To solve the problem, the genetic coding with a better fitness will be reserved for the evolution of succeeding generations.
- Improvement in the mutation mechanism will help solve the assembly problem.

3. RESEARCH METHOD

Solving SALBP-E using a genetic algorithm with partitioned chromosome was done through the following steps:

1) Develop a precedence diagram which describing assembly line problem will be solved.
2) Make partitions that divide the network diagram from left to right into several regions as done in the Region Approach Method (Bedworth and Bailey, 1987). A difference is that in this method all the tasks were assigned as early as possible in the feasible region to ensure that genetic operation does not violated the precedence constraints, while in Region Approach all tasks are assigned to the latest precedence region possible to ensure that task with few dependencies will be considered for assignment at the end of the schedule.
3) Build initial chromosome which represented by real number encoding (Sivanandam dan Deepa, 2008). The resulting chromosome consists of 42 genes which describes the number of tasks (elements) contained in the cooling lines. In this case were formed 19 regions, where the member of first region are all free elements without

predecessors, and member of 2nd region are all elements follower of region 1. The member of 3rd region and so on were determined in the same manner as before. Figure 1 shows precedence diagram and in Table 1 can be seen the distribution of genes (tasks) within each region in initial chromosome.

4) Generate the initial population (1st generation), which consists of a number of chromosomes. Chromosomes obtained by randomizing the genes contained in each region (Figure 2). By limiting randomization only for genes within each region, it is expected the obtained solution does not violate the existing network precedence.
5) The placement all genes of each chromosome to each station according to the maximum workstation cycle time established before, and based on the sequence obtained from the randomization. All tasks are placed so it didn't break the existing precedence constraint.
6) Calculate the fitness value of the chromosomes in the initial population. Fitness value is determined by the following formula: :

$$Fitness\ (F) = Max\ E = \frac{\sum_{i=1}^{M} T_i}{(C)(M)} \times 100 \quad (8)$$

7) Construct a new population
   The construction procedure of a new population is divided into 3 steps, ie selection, crossover, and mutation. Selection is choosing chromosomes that will be the new population. Selection of chromosomes can be done using the roulette wheel or tournament procedure.
   Selection of chromosomes for crossover is determined by comparing the random numbers value of a chromosome with a crossover probability (P_c). If the random numbers<P_c then the chromosome is crossed over, but if the random number>P_c then the chromosomes are not crossed over.
   This case used a Position-Based Crossover procedure to generate offsprings that will replace the parent chromosomes in the next generation. Figure 3 shows how 2 chromosomes are crossed over into 2 offsprings.
8) The resulting offsprings are checked whether they are feasible and does not break the precedence constraint.
9) With a mutation probability, mutate new offsprings use shift mutation procedure. Examples of mutated offspring can be seen at Figure 4.
10) Compute and evaluate the fitness value of offsprings produced. Best fitness value indicated by the highest line efficiency.
11) Elitism
    In the elitism process the fitness value of each chromosome are sorted to determine the chromosome with the best value. The best chromosome or the few best chromosomes are copied to the new population to prevent damage caused by a genetic process (Suyanto 2005; Sivanandam and Deepa, 2008).
12) After the elitism, then restarted the process of establishing a new population that began from chromosome selection until crossover and mutation to generate the next generation. The process stops when it reached the specified number of generations.
13) Calculating production output produced by the chromosome with the highest fitness value in the last generation to see wether the production target (the goal of this research) is achieved. Production output (O) is calculated by dividing the available working time per day with a maximum cycle time of work stations (Ti max).

$$O = \frac{Available\ time\ per\ day}{(T_i)max} \quad (8)$$

4. RESULT AND DISCUSSION

From Table 2 it can be observed the comparison of assembly line performance before and after rebalancing using GA. These results were obtained with the help of Borland Delphi program. In this case, the crossover probability is set at 0,25, and the mutation probability, 0,01. The best result was obtained by setting the maximum number of generation, 10.

Figure 5 shows the input processes which run on the Borland Delphi program, that are the maximum cycle time work stations,

crossover probability, mutation probability and the maximum number of generation.

Tabel 2. Assembly line performance, before and after rebalancing

Assembly Line Performance	Initial condition	*After rebalancing with* GA approach
Line Efficiency (%)	52.46	91.42
Smoothness Index	75.35	13.25
Max workstation time (second)	36.60	28.64
The number of workstations	15	11
Production Output (pcs/day)	786	1005
Production target (pcs/day)	1000	

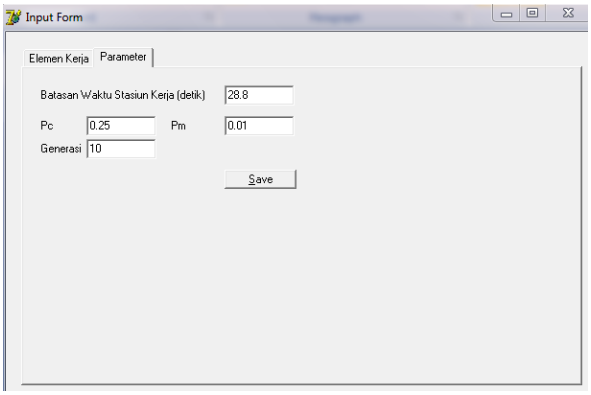


Figure 5. Input Program GA

Figure 6 illustrate working station 1 to 4 resulted, include the tasks (elements) and their standard time. Figure 6 also presented assembly line performance which indicated by maximum workstations time, line efficiency and smoothness index.

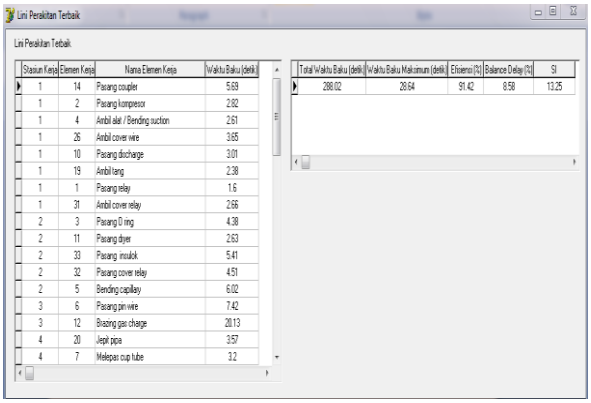
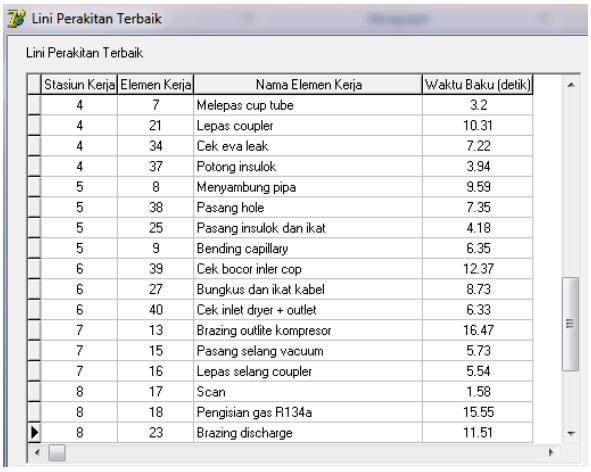


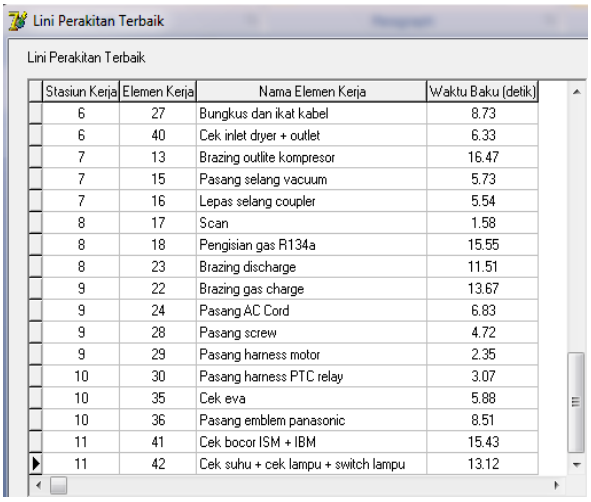
Figure 6. Stasiun 1 sampai 4 dan Performansi Lini

Figure 7 illustrates station 4 to 8 include task elements and standard time of each element. In Figure 8 it can be seen workstations 6 through 11.



Stasiun Kerja	Elemen Kerja	Nama Elemen Kerja	Waktu Baku (detik)
4	7	Melepas cup tube	3.2
4	21	Lepas coupler	10.31
4	34	Cek eva leak	7.22
4	37	Potong insulok	3.94
5	8	Menyambung pipa	9.59
5	38	Pasang hole	7.35
5	25	Pasang insulok dan ikat	4.18
5	9	Bending capillary	6.35
6	39	Cek bocor inler cop	12.37
6	27	Bungkus dan ikat kabel	8.73
6	40	Cek inlet dryer + outlet	6.33
7	13	Brazing outlite kompresor	16.47
7	15	Pasang selang vacuum	5.73
7	16	Lepas selang coupler	5.54
8	17	Scan	1.58
8	18	Pengisian gas R134a	15.55
8	23	Brazing discharge	11.51

Figure 7. Stasiun 4 sampai 8



Stasiun Kerja	Elemen Kerja	Nama Elemen Kerja	Waktu Baku (detik)
6	27	Bungkus dan ikat kabel	8.73
6	40	Cek inlet dryer + outlet	6.33
7	13	Brazing outlite kompresor	16.47
7	15	Pasang selang vacuum	5.73
7	16	Lepas selang coupler	5.54
8	17	Scan	1.58
8	18	Pengisian gas R134a	15.55
8	23	Brazing discharge	11.51
9	22	Brazing gas charge	13.67
9	24	Pasang AC Cord	6.83
9	28	Pasang screw	4.72
9	29	Pasang harness motor	2.35
10	30	Pasang harness PTC relay	3.07
10	35	Cek eva	5.88
10	36	Pasang emblem panasonic	8.51
11	41	Cek bocor ISM + IBM	15.43
11	42	Cek suhu + cek lampu + switch lampu	13.12

Figure 8. Stasiun 6 sampai 11

Improved line efficiency presented in Table 2 and Figure 6 demonstrate that there was an increasing in the utilization of generated workstation. It means also there was a reduction at work stations idle time. A decline in the value of smoothness index indicates that there was a more equal distribution of workload among each station.

Table 1 also indicate that there is an increase in production output generated on a rebalanced assembly line. The new production output is calculated as follows :

$$O = \frac{28{,}800\ second}{28{,}64\ second/pcs} = 1005\ pcs$$

5. CONCLUSION

Solving simple assembly line balancing problem Type-E using GA with partitioned

chromosomes can increase the line efficiency, decrease the smoothness index and increase production output by 27.86%. In this case GA with partitioned chromosome have improved genetic process performance through the formation more appropriate initial chromosome. The results also show that the satisfactory fitness value can be achieved in a reasonably short iteration process, ie the generation of 10, and a production target of 1000 pcs was reached.

6. REFERENCES

(a) Becker, C. and Scholl, A. (2006). A survey on problems and methods in generalized assembly line balancing. *European Journal of Operational Research*, Vol. 168, 694-715.
(b) Boysen, N., Fliedner, M. and Scholl, A. (2007). A classification of assembly line balancing problems. *European Journal of Operational Research*, Vol. 183, 674–693.
(c) Chase, R.B., Aquilano, N.J., and Jacobs, F.R. (1998). *Production and Operations Management,* Manufacturing and Services. 8th ed., Irwin-McGraw-Hill.
(d) Chong, K.E., Omar, M.K. and Bakar, N.A. (2008). Solving Assembly Line Balancing Problem using Genetic Algorithm with Heuristics-Treated Initial Population. *Proceedings of the World Congress on Engineering (WCE)* 2008 Vol II, July 2 - 4, 2008, London, U.K.
(e) Kriengkorakot, N and Pianthong, N. (2007). The Assembly Line Balancing Problem : Review articles. *KKU Engineering Journal*, Vol. 34 No.2 , 133 – 140, March – April 2007.
(f) Rashid, M.F.F., Hutabarat, W., and Tiwari, A. (2011). A Review on Assembly Sequence Planning and Assembly Line Balancing Optimisation using Soft Computing Approaches. *International Journal of Advanced Manufacturing Technology*, 2011, Vol. 59 (1-4), 335-349.
(g) Scholl, A. and Becker, C., 2006. State-of-the-art Exact and Heuristic Solution Procedures for Simple Assembly Line Balancing. *European Journal of Operational Research*. Vol. 168, Issue 3: pp. 666-693.
(h) Sivanandam SN, Deepa SN. 2008. *Introduction to Genetic Algorithms*. Berlin: Springer-Verlag
(i) Suwannarongsri, S and Puangdownreong, D. (2008) Optimal assembly line balancing using tabu search with partial random permutation technique. *International Journal of Management Science and Engineering Management*, Vol. 3 No. 1, pp. 3-18. ISSN 1750-9653, England, UK.
(j) Suyanto. (2005). *Algoritma Genetika Dalam Matlab.* Penerbit Andi, Yogyakarta.
(k) Tseng, H.E. (2006). Guided genetic algorithms for solving a larger constraint assembly problem. *International Journal of Production Research*, Vol. 44, No. 3, 601–625

AUTHOR BIOGRAPHIES

Nora Azmi is a lecturer in Department of Industrial Engineering, Faculty of Industrial Technology, Trisakti University, Jakarta. She obtained her doctoral degree from Agroindustrial Engineering Department, Bogor Agricultural Institute and master degree from Industrial Engineering Department, Bandung Institute Technology Her research interests are in the area of Production Planning & Control, Genetic Algorithm and Ergonomics. Her email address is noraazmi@yahoo.com.

Iman Yahya Azzubaidi completed his undergraduate study at Industrial Engineering Department, Trisakti University in 2012. In the last 2 years of her studies she joined as an lectures assistant in the Production System Laboratory at Industrial Engineering Department. Her research area are in assembly line balancing problem.

Sumiharni Batubara is a lecturer in the Industrial Engineering Department, Trisakti University, Jakarta, since 1993. She received her Master degree in Industrial Engineering from Institut Technology Bandung. Her research interest are in the area of Production Planning & Control, Inventory Control and Just In Time. Now, she is a head of the Production System Laboratory, Industrial Engineering

Department, Trisakti University. She is also a member of the Society of Industrial Engineers.

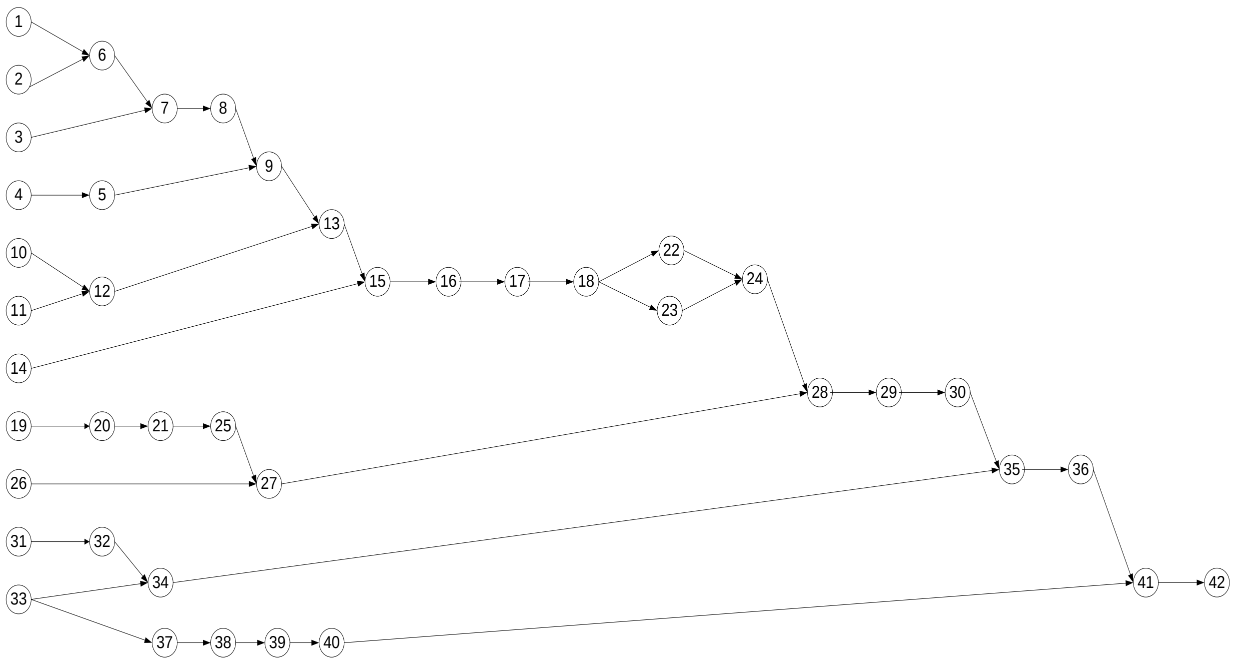


Figure 1. Precedence Diagram

Table 1. Division of Region at Initial Chromosome

Region	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Tasks	1	6	7	8	9	13	15	16	17	18	22	24	28	29	30	35	36	41	42
	2	5	21	25	27	40					23								
	3	12	34	38	39														
	4	20	37																
	10	32																	
	11																		
	14																		
	19																		
	26																		
	31																		
	33																		

Kr 1	2	3	11	4	1	10	31	14	19	33	26	6	5	12	20	32	7	21	34	37	8	25	38	9	27	39	13	40	15	16	17	18	22	23	24	28	29	30	35	36	41	42
Kr 2	33	2	31	4	3	10	14	19	26	11	1	6	5	32	20	12	7	21	37	34	38	25	8	39	27	9	40	13	15	16	17	18	23	22	24	28	29	30	35	36	41	42
Kr 3	2	26	10	14	31	4	3	33	19	11	1	12	32	6	5	20	21	34	37	7	25	8	38	27	9	39	40	13	15	16	17	18	22	23	24	28	29	30	35	36	41	42
Kr 4	1	2	26	3	11	14	19	31	4	33	10	20	5	12	6	32	7	21	34	37	38	25	8	39	27	9	40	13	15	16	17	18	23	22	24	28	29	30	35	36	41	42
Kr 5	31	33	2	14	3	1	4	26	19	11	10	6	32	12	5	20	7	37	34	21	8	25	38	9	27	39	13	40	15	16	17	18	22	23	24	28	29	30	35	36	41	42

Figure 2. Five chromosomes of initial population

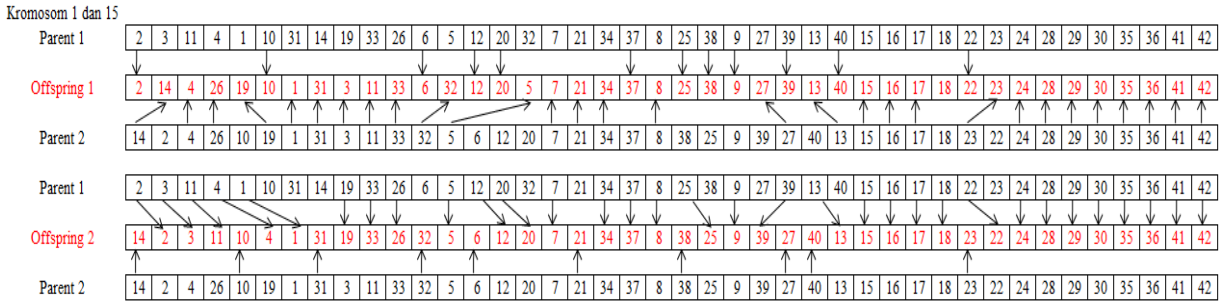


Figure 3. Example of Cross Over Procedure using Position Based Crossover

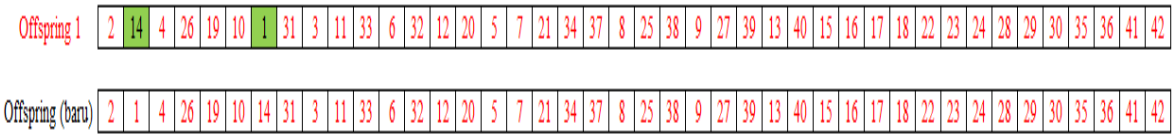


Figure 4. Example of Shift Mutation Procedure