International Journal of Performability Engineering Vol. 7 No. 3, May 2011, pp. 263-277.

Printed in India

Optimal Assignment of a Two Types of Components in a Multi-state Stochastic-flow Network

YI-KUEI LIN* and CHENG-TA YEH

Department of Industrial Management
National Taiwan University of Science & Technology
Taipei, Taiwan 106, R.O.C.

(Received on March 17, 2010 and revised on December 18, 2010)

Abstract: A real-life system is usually modeled to be a network with edges and nodes for the performance evaluation. This paper devotes to finding out the optimal two-type component assignment with maximal system reliability, in which the components are separated to be two sets: one assigned to edges and the other set assigned to nodes. Each component should be multistate due to complete failure, partial failure, maintenance, etc. Such a system according to a two-type component assignment is a stochastic-flow network. Furthermore, each component has a transmission cost in practice. Therefore, the system reliability is the probability that d units of demand are transmitted through the network successfully subject to a budget. A genetic algorithm based algorithm is developed to solve the proposed problem in which the system reliability according to a two-type component assignment is evaluated in terms of minimal paths and Recursive Sum of Disjoint Products. The experimental results show that the proposed algorithm can be executed in reasonable time.

***Keywords*:** System reliability; Two-type component assignment; Stochastic-flow network; Transmission Cost; Genetic algorithm

1. Introduction

The assignment problem (AP) focused on searching for the optimal assignment from a set of components to a set of locations of a system. The component may be a communication device of computer system, a mechanism of manufacturing system, etc. According to the studies of Pentico [1] and Winston [2], many APs focused on minimizing total cost or maximizing total profit and only involved the assignment single-type component assignment. In addition, any system in such APs was deterministic since the state of the component assigned to the system was given.

A real-life system is usually modeled to be a network with edges and nodes, in which each component (edge/node) is multistate due to complete failure, partial failure, maintenance, etc. Such a network is namely a stochastic-flow network (SFN) [3-21]. The system reliability of a SFN is an important performance index of a system. Several studies have evaluated the system reliability for a SFN in terms of minimal paths (MPs) [3-5] or minimal cuts (MCs) [3,4,6]. The system reliability of a SFN with failed nodes is evaluated in terms of MPs [7] and MCs [8]. For practical applications, the system reliability is evaluated subject to a budget [9-11,13,14]. For the system reliability optimization problems, several research mainly discussed the network structure problems [15], the flow assignment problems [16, 17], commodity allocation problems [18-20], and the stochastic network interdiction problem [21].

*Corresponding authors email: yklin@mail.ntust.edu.tw

From the viewpoint of quality management, one confronted problem is to find out the optimal component assignment with maximal system reliability, in which each component is multistate and can be assigned to an edge or node. Lin and Yeh's previous paper [22] has explored the optimal component assignment with maximal system reliability for SFN under a budget constraint. However, it only considered the single type of component. This study considers the two-type component assignment problem, in which the components are separated to two sets: one set of components mainly assigned to the edges and the other set of components mainly assigned to nodes. For example of a computer network, a transmission line (resp. a transmission facility) is the component which can be assigned to exact one edge (resp. node). Each transmission line (resp. a transmission facility) is combined with several physical lines (resp. a transmission devices) such as fiber cables, twisted pairs or coaxial cables (resp. hubs, routers, switches or bridges) all of which have a capacity or may be failed with a probability. Moreover, since the flow traveling through the network results in a transmission cost, the system reliability is defined as the probability that d units of demand are successfully transmitted from an origin O to a destination D under a budget constraint in this study.

Synthesizing the above statement, the discussed problem that is to maximize the system reliability according to the optimal two-type component assignment under a budget constraint is called the SRDCAB problem for short. The system reliability evaluation of a SFN is an NP-hard problem [23] and the APs are also NP-hard [24]. Since the SRDCAB problem combines the characteristics of the SFN and the APs, it is NP-hard. A genetic algorithm (GA) has the ability to find the better optimal solutions than other heuristic algorithms for assignment problems [25,26]. Intuitively, the problem can be solved by the implicit enumeration method, but it is time-consuming if the system is large. Hence, we propose a genetic algorithm (GA) based algorithm to solve the SRDCAB problem, and call it SRDCAB-GA for short. Another reason is that GA is a robust search technology and has been widely applied to many APs [25-28]. In the proposed algorithm, a two-type component assignment is designated as a chromosome, and the fitness function combines the MPs and Recursive Sum of Disjoint Products (RSDP) to evaluate the system reliability. Through this algorithm, the optimal two-type component assignment with maximal system reliability can be found in reasonable time.

The rest of this paper is organized as follows. The assumptions are shown in Section 2. The SRDCAB problem is explored in Section 3. Section 4 subsequently develops the SRDCAB-GA algorithm to solve the proposed problem. One large and two common-used networks are demonstrated in Section 5 to illustrate the proposed algorithm. Conclusions are finally described in Section 6, along with recommendations for future research.

2. Assumptions

Let $\Psi = (\boldsymbol{E}, \boldsymbol{V})$ be a network with a single origin O and a single destination D, where $\boldsymbol{E} = \{e_i | 1 \le i \le n\}$ denotes the set of n edges connecting nodes and $\boldsymbol{V} = \{e_i | n + 1 \le i \le n + q\}$ denotes the set of q nodes except for O and D. The total m MPs in Ψ are designated as mp_1, mp_2, …, mp_m. The set $\boldsymbol{\Gamma_1} = \{\pi_s | 1 \le s \le \omega_1\}$ denotes the set of ω_1 Type 1 components which are ready to be assigned to the edges and the set $\boldsymbol{\Gamma_2} = \{\pi_s \, | \omega_1 + 1 \le s \le \omega_1 + \omega_2\}$ denotes the set of ω_2 Type 2 components which are ready to be assigned to the nodes. The cost per unit of data transmitted through component π_s is designated as c_s. Each component π_s has multiple capacities, $0 = k_s(1) < k_s(2) < \ldots < k_s(M_s)$, where $k_s(b)$ is the bth capacity of component π_s for $b = 1, 2, \ldots, M_s$, and $k_s(M_s)$ is the maximal capacity of π_s. Let $X = (x_1, x_2, \ldots, x_{n+q})$ be a two-type component assignment. Let $Y = (y_1, y_2, \ldots, y_{n+q})$ be a current capacity vector where y_i denotes the current capacity of e_i for all i, and $F = (f_1, f_2, \ldots, f_m)$ be a flow vector where f_j denotes the flow through mp_j, $j = 1, 2, \ldots, m$. Several assumptions are listed as follows.

Assumptions:

I. Each network is a terminal-pair network.

II. Each Type 1 component should be assigned to at most one edge and each edge must contain exact one Type 1 component.

III. Each Type 2 component should be assigned to at most one node and each node must contain exact one Type 2 component.

IV. The capacities of different Type 1 component and Type 2 component are statistically independent.

V. Flow in a network must satisfy the flow-conservation law [29].

Definition 1: A path is a set of edges and nodes which connects O and D.

Definition 2: An MP is a path whose proper subsets are no longer paths.

3. Problem of Stochastic-flow Network according to Two-type Component Assignment

3.1 Flow model

The maximal capacity of e_i denoted by $k_{x_i}(M_{x_i})$ is equal to $k_s(M_s)$ according to the component assignment X. Thus, the maximal capacity according to X is $(k_{x_1}(M_{x_1}), k_{x_2}(M_{x_2}), \ldots, k_{x_{n+q}}(M_{x_{n+q}}))$ designated as $\boldsymbol{M}_X$. Any capacity vector $Y \in \mathbf{Y}_X$ that is said to be feasible according to X must satisfy the following constraint:

$$y_i \le k_{x_i}(M_{x_i}) \text{ for } i = 1, 2, \ldots, n + q, \tag{1}$$

Constraint (1) means the current capacity y_i cannot exceed the maximal capacity of e_i according to X. A feasible F under $\boldsymbol{M}_X$ is said to satisfy the following constraints:

$$f_j \le \min\{k_{x_i}(M_{x_i}) \mid i : e_i \in mp_j\} \qquad j = 1, 2, \ldots, m, \text{ and} \tag{2}$$

$$\sum (f_j \mid j : e_i \in mp_j) \le k_{x_i}(M_{x_i}) \qquad i = 1, 2, \ldots, n + q, \tag{3}$$

where $\min\{k_{x_i}(M_{x_i}) \mid i : e_i \in mp_j\}$ represents the maximal capacity of mp_j, and $\sum (f_j \mid j : e_i \in mp_j)$ means the total flow traveling through e_i. Constraint (2) means the flow through mp_j should not exceed the maximal capacity of mp_j. Constraint (3) means the total flow through e_i should not exceed the maximal capacity of e_i according to X. Particularly, any F satisfies constraint (3) also satisfies constraint (2). For convenience, let $F_X = \{F \mid F$ satisfies constraint (3)$\}$ be the set of F feasible under $\boldsymbol{M}_X$. Similarly, any feasible F under $Y \in \mathbf{Y}_X$ is said to satisfy the following constraint:

$$\sum \{f_j \mid j : e_i \in mp_j\} \le y_i \qquad i = 1, 2, \ldots, n + q. \tag{4}$$

Let $\Omega_Y = \{F \mid F$ meets constraint (4)$\}$ denote the set of F feasible under Y. The maximal flow of Ψ under Y is designated as $V(Y)$, and thus $V(Y) \equiv \max\{\sum_{j=1}^{m} f_j \mid F \in \Omega_Y\}$.

3.2 Transmission Cost and System Reliability Definition

The transmission cost is affected by the total flow through e_i. Any capacity vector Y satisfies both $V(Y) \geq d$ and budget C according to X if there exists a $F \in \Omega_Y$ such that

$$\sum_{j=1}^{m} f_j \geq d \text{ and} \tag{5}$$

$$\sum_{i=1}^{n+q}\left(c_{x_i}\sum_{j:e_i \in mp_j} f_j\right) \leq C. \tag{6}$$

Constraint (5) says the amount of flow from O to D meets the demand d. Constraint (6) describes the total transmission cost should not exceed budget C, where $c_{x_i}\sum_{j:e_i \in mp_j} f_j$ means the total cost of the total flow transmitted through e_i.

Let $\hat{\mathbf{Y}}_X = \{Y \mid$ there exists an $F \in \Omega_Y$ satisfying both constraints (5) and (6)$\}$ is the set of Y such that $V(Y) \geq d$ under budget C according to X. Then, the system reliability denoted by $R_{d,C}(X)$ is defined as the probability that the maximal flow according to X is no less than a given demand d subject to C, i.e., $R_{d,C}(X) \equiv \Pr\{Y|Y \in \hat{\mathbf{Y}}_X\}$.

3.3 Generate all (d, C)-MP

It is not an efficient way to enumerate all $Y \in \hat{\mathbf{Y}}_X$ [5,7]. Thus, A (d, C)-MP is defined to improve the efficiency in system reliability computation.

Definition 3: The minimal capacity vector Y in $\hat{\mathbf{Y}}_X$ such that $U \notin \hat{\mathbf{Y}}_X$ for any capacity vector $U < Y$ is a (d, C)-MP (where $Y \leq U$ if and only if $y_i \leq u_i$ for each i and $Y < U$, if and only if $Y \leq U$ and $y_i < u_i$ for at least one i).

Suppose $Y_1, Y_2, \ldots, Y_v$ are v (d, C)-MPs, and thus $\hat{\mathbf{Y}}_X = \{\{\bigcup_{i=1}^{v}\{Y \mid Y \geq Y_i\}\}$. Then, $R_{d,C}(X) = \Pr\{Y|Y \in \hat{\mathbf{Y}}_X\} = \Pr\{\bigcup_{i=1}^{v}\{Y \mid Y \geq Y_i\}\}$. A flow vector F is said to satisfy the exact demand subject to C if $F \in F_X$ satisfies constraints (6) and (7),

$$\sum_{j=1}^{m} f_j = d, \tag{7}$$

Let $\mathbf{F} = \{F|F \in F_X$ satisfies constraints (6) and (7)$\}$ be the set of F satisfying d and C under $\boldsymbol{M}_X$. If $Y \in \hat{\mathbf{Y}}_X$ is a (d, C)-MP, then there exists an feasible $F \in \mathbf{F}$ under Y such that

$$y_i = k_{x_i}(b_i) \quad \textit{where } b_i \in \{1, 2, \ldots, M_{x_i}\} \textit{ such that } k_{x_i}(b_i - 1) < \sum\{f_j \mid j : e_i \in mp_j\} \leq k_{x_i}(b_i),\ i = 1, 2, \ldots, n + q. \tag{8}$$

Any $Y = (y_1, y_2, \ldots, y_{n+q})$ transformed from $F \in \mathbf{F}$ through Eq. (8) satisfies $V(Y) \geq d$ and C, and is treated as a (d, C)-MP candidate based on Lemma 1. Let $Y_1, Y_2, \ldots, Y_w$ be such (d, C)-MP candidates and each (d, C)-MP candidate is further checked whether it is a (d, C)-MP or not by the following algorithm.

Algorithm I: //Identify which $Y_1, Y_2, \ldots, Y_w$ are (d, C)-MPs.

1) $I = \varnothing$ (I is the stack which stores the index of non-(d, C)-MP Y. Initially, I is empty.)
2) For $i = 1$ to w & $i \notin I$.
3) For $j = i + 1$ to w, & $j \notin I$.

4) If $Y_j < Y_i$, then Y_i is not a (d, C)-MP, $I = I \cup \{i\}$, and go to step 7.
   Else if $Y_j \geq Y_i$, then Y_j is not a (d, C)-MP, $I = I \cup \{j\}$.
5) Next j.
6) Y_i is a (d, C)-MP.
7) Next i.

3.4 System Reliability Evaluation

The system reliability according to X, $R_{d,C}(X) = \Pr\{\bigcup_{i=1}^{v}\{Y \mid Y \geq Y_i\}\}$, is calculated in terms of the RSDP [30]. The RSDP proposed by Zuo *et al*. [30] for system reliability evaluation is based on Sum of Disjoint Products (SDP) principle and a special maximum operator, "⊕". The special maximum operator is defined as follows:

Definition 4: A special maximum operator, "⊕", is defined as

$$Y_1 \oplus Y_2 = (\max(y_{1i}, y_{2i})) \qquad \text{for } i = 1, 2, \ldots, n + q. \tag{9}$$

For example, if Y_1 = (1, 3, 1, 3, 4) and Y_2 = (2, 2, 0, 1, 4), $Y_1 \oplus Y_2$ = (max(1, 2), max(3, 2), max(1, 0), max(3, 1) , max(4, 4)) = (2, 3, 1, 3, 4).

3.5 Problem Formulation

The mathematical programming formulation for the SRDCAB problem is shown as follows:

$$\textit{Maximize } R_{d,C}(X) = \Pr\{\bigcup_{i=1}^{v}\{Y \mid Y \geq Y_i\}\} \tag{10}$$

Subject to:

$$x_i = s \qquad \pi_s \in \boldsymbol{\Gamma_1} \text{ for } i = 1, 2, \ldots, n, \tag{11}$$

$$x_i = t \qquad \pi_t \in \boldsymbol{\Gamma_2} \text{ for } i = n+1, n+2, \ldots, n+q, \tag{12}$$

$$x_i \neq x \qquad \text{for } i \neq j \text{ and } i, j = 1, 2, \ldots, n, \text{ and} \tag{13}$$

$$x_i \neq x_j \qquad \text{for } i \neq j \text{ and } i, j = n+1, n+2, \ldots, n+q. \tag{14}$$

Constraints (11) - (14) are utilized to meet assumptions I and II. Therefore, the component assignment X that satisfies constraints (11) - (14) is said to be feasible. Note that $R_{d,C}(X) = 0$ if $\hat{\mathbf{Y}}_X = \varnothing$. The optimal component assignment with maximal system reliability is computed by maximizing objective function (10).

4. SRDCAB-GA Development

Figure 1 illustrates the entire process of the proposed algorithm. Several major parameters, population size (designated as λ), crossover rate (designated as δ), mutation rate (designated as γ) and number of generation (designated as θ), must be defined. The SRDCAB-GA starts with an initial population including of λ chromosomes, and then evaluate each chromosome (solution) of the population by a fitness function. The optimal solution is improved from one generation to another by the evolution process composed of selection, crossover, and mutation operators. The terminal condition is responsible to terminate the SRDCAB-GA algorithm. In this study, SRDCAB-GA stops as it runs for θ generations, and then returns the optimal solution of the current generation.

4.1 Chromosome Coding

We adopt the integer coding to represent a chromosome. Such a chromosome should conform to a two-type component assignment. Hence, a chromosome is denoted as $G = (g_1, g_2, \ldots, g_{n+q})$. If Type 1 component s is assigned to edge e_i, $g_i = s$ and $s \in \boldsymbol{\Gamma_1}$; if Type 2 component s is assigned to node e_j, $g_j = s$ and $s \in \boldsymbol{\Gamma_2}$.

4.2 Fitness Function

The fitness function is used to evaluate system reliability for each chromosome. That is, the system reliability of a chromosome is its fitness value. In the SRDCAB-GA, if chromosome G results in $\hat{\mathbf{Y}}_G = \varnothing$ according to G, then $R_{d,C}(G)$ is equal to a given penalty value which must be very small. The following algorithm evaluates the system reliability of each chromosome.

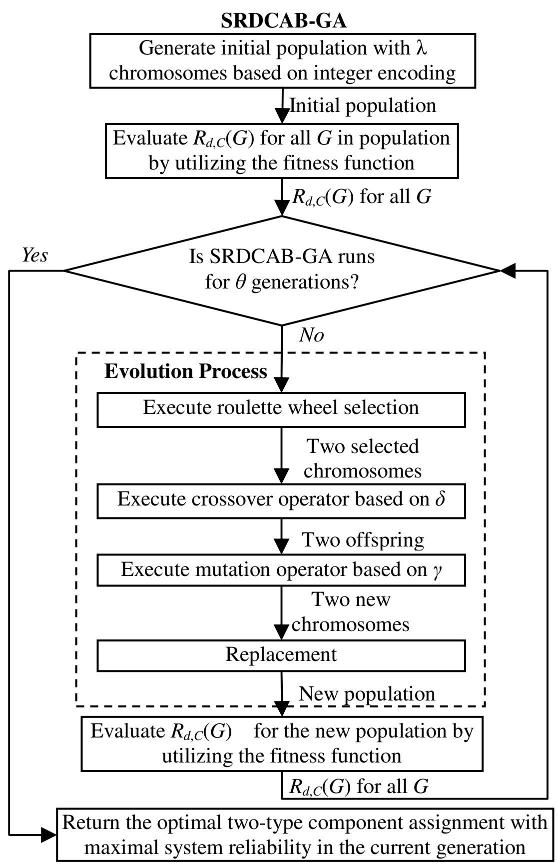

Figure 1: The process of SRDCAB-GA

Algorithm II: // Evaluate system reliability

Step 1. Find all F satisfying the following constraints:

$$\sum\{f_j \mid j : e_i \in mp_j\} \le k_{g_i}(M_{g_i}) \qquad i = 1, 2, \ldots, n+q, \tag{15}$$

$$\sum_{i=1}^{n+q}\left(c_{g_i}\sum\nolimits_{j:e_i\in mp_j} f_j\right) \le C \text{ and,} \tag{16}$$

$$\sum_{j=1}^{m} f_j = d. \tag{17}$$

If no feasible F exists, a penalty value is assigned to be $R_{d,C}(G)$, and then repeat this algorithm to evaluate the next chromosome.

Step 2. Transform each F into Y through the following equation:

$$y_i = k_{g_i}(b_i) \text{ where } b_i \in \{1, 2, \ldots, M_{g_i}\} \text{ such that } k_{g_i}(b_i - 1) < \sum\{f_j \mid j : e_i \in mp_j\} \le k_{g_i}(b_i),\quad i = 1, 2, \ldots, n+q. \tag{18}$$

Step 3. Utilize Algorithm I to obtain all (d, C)-MPs from all Y transformed from step 2.

Step 4. Suppose all (d, C)-MPs are Y_1, Y_2, …, Y_v. Calculate $R_{d,C}(G) = \Pr\{\bigcup_{i=1}^{v}\{Y \mid Y \ge Y_i\}\}$ by the RSDP.

4.3 Evolution Process

At first, the roulette wheel selection is implemented twice to generate the parental chromosomes. Our proposed crossover operator is based on the ideal of the single-point crossover. Suppose that two chromosomes, (1, 3, 2, 5, 6, 4, **11**, **15**) and (5, 6, 1, 3, 4, 2, **12**, **14**), are selected from the population, in which the bold words represent Type 2 components. The crossover operator should be executed for the part of $g_i = (g_i|1 \le i \le n)$ and for the part of $g_i = (g_i|n+1 \le i \le n+q)$. Firstly for $g_i = (g_i|1 \le i \le n)$, a crossover point ($CP$) is randomly generated to be g_5, and then two chromosomes exchange with each other at the right hand side of CP including of CP to be (1, 3, 2, 5, $\underline{4}$, $\underline{2}$, **11**, **15**) and (5, 6, 1, 3, $\underline{6}$, $\underline{4}$, **12**, **14**). After this work, the g_3 and g_6 in first chromosome have the same value, as are the g_2 and g_5 in the second chromosome. For first (resp. second) chromosome, g_3 (resp. g_2) is randomly changed as the gene on the left hand side of CP from the second (resp. first) chromosome which does not appear in the part of $g_i = (g_i|1 \le i \le n)$ of the first (resp. second) chromosome. In other words, g_3 in the first chromosome can be only changed as g_2 from the second chromosome. Similarly, g_2 in the second chromosome is only changed as the g_3 gene from the first chromosome. Then, the two chromosomes are (1, 3, 6, 5, 4, 2, **11**, **15**) and (5, 2, 1, 3, 6, 4, **12**, **14**). The same process of the crossover is executed for the part of $g_i = (g_i|n+1 \le i \le n+q)$ and then obtains (1, 3, 6, 5, 4, 2, **11**, **14**) and (5, 2, 1, 3, 6, 4, **12**, **15**) as $CP = g_8$.

The proposed mutation should also be executed for the part of $g_i = (g_i|1 \le i \le n)$ and the part of $g_i = (g_i|n+1 \le i \le n+q)$. For example of chromosome (1, 3, 6, 5, 4, 2, **11**, **14**), a random mutation point (MuP) is created to be g_4. If the value of g_4 becomes 1, then g_4 exchanges with g_1 to avoid that g_1 and g_4 have the same value 1, and then the chromosome becomes ($\underline{5}$, 3, 6, $\underline{1}$, 4, 2, **11**, 1**4**). If the value of g_4 becomes 7, then the chromosome directly becomes (5, 3, 6, $\underline{7}$, 4, 2, **11**, **14**). Similarly for the part of $g_i = (g_i|n+1 \le i \le n+q)$, the chromosome becomes (5, 3, 6, 7, 4, 2, <u>**18**</u>, **14**) if MuP is supposed to be g_7. Through our proposed crossover and mutation, no duplicated gene value appears in the same chromosome.

5. Numerical Experiment

A large random computer network and two common-used computer networks are taken to be discussed in this section. In this case, each Type 1 component is combined with several OC-18 (Optical Carrier 18) lines, and each Type 2 component is composed of several hubs. Each OC-18 line provides two capacities, 0 bps (bit per second) and approximate 1Gbps (Giga bit per second). Each hub owns two capacities, 4Gbps if the hub works and 0 bps if the hub fails. Since all components are provided by various suppliers, they have different capacities and probability distributions. The proposed algorithm is programmed in MATLAB programming language and executed on a personal computer with Intel Core 2 Quad CPU 2.4G and 2G RAM.

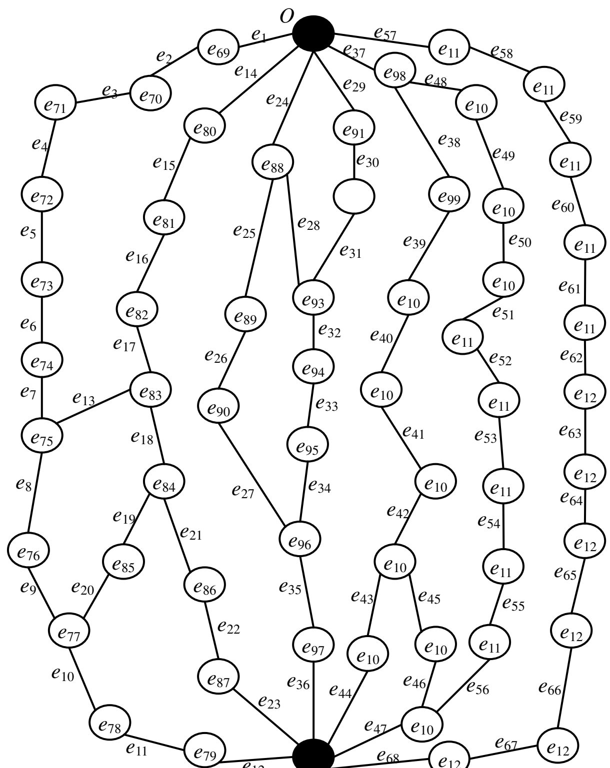

Figure 2: A large random computer network

A random large computer network (shown in Figure 2) with 68 edges, 57 nodes and 17 MPs and two common-used computer networks – the simple computer network [6] and the OCT network [31] are adopted to demonstrate the practicability of the proposed algorithm, in which the simple network (shown in Figure 3) includes 5 edges, 2 nodes and 4 MPs, and the OCT network (shown in Figure 4) includes 29 edges, 24 nodes and 9 MPs.

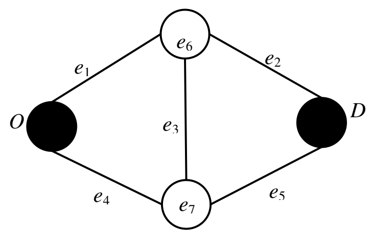

Figure 3: A simple computer network [6].

We utilize the data in Table 1 (refer to Appendix) to examine the computational efficiency of the SRDCAB-GA for this case. The SRDCAB-GA is implemented with λ = 50, δ = 0.7, γ = 0.1 and θ = 4500. Particularly, we consider a termination constraint – the SRDCAB-GA stops if there is no improvement of the maximal system reliability over the next 400 generations. Firstly, we are aimed at the random large network. Under d = 5 Gb and C = 3000 (New Taiwan Dollar, NTD) (resp. d = 10 Gb and C = 5400), Table (a) (resp. (b)) (refer to Appendix) shows that the optimal two-type component assignment has maximal system reliability 0.9896641209 (resp. 0.8136644097).

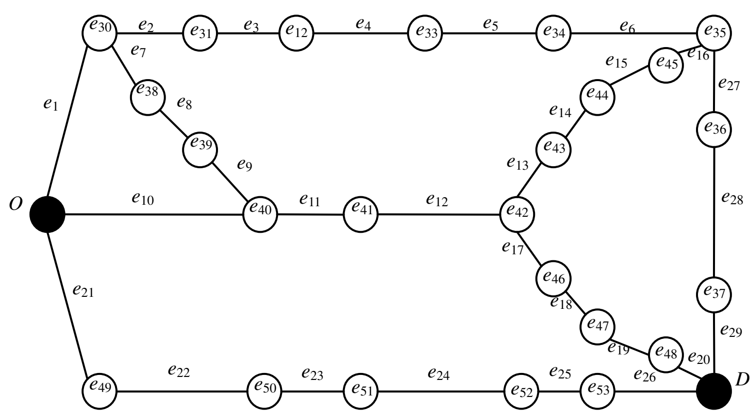

Figure 4: The topology of OCT network [31].

Figure 5 illustrates the progress of searching for the optimal solution. The optimal solution is convergent at the 2457th (resp. 3734th) generation and the algorithm stops at the 2856th (resp. 4134th) generation as d = 5 Gb and C = 3000 (resp. d = 10 Gb and C = 5400). Obviously, the optimal solution is convergent in advance as smaller d. Furthermore, the two experiments are accomplished in approximately 86 and 124 hrs, respectively. Subsequently, we execute the experiments for the other two networks using the SRDCAB-GA with the same parameters but the termination constraint is that the algorithm stops if there is no improvement of the maximal system reliability over the next 300 generations. When d = 5 and C = 100 (resp. d = 5 and C = 1700) for the simple network (resp. the OCT network), the maximal system reliability is 0.9663200106 (resp. 0.9822438592) and the CPU time is 2 (resp. 496) seconds.

Significantly, the proposed algorithm takes more time in searching the optimal solution with maximal system reliability for the more complex network. Figure 6 illustrates that the optimal solution is convergent at the 251st (resp. 3001st) generation and the algorithm stops at the 551st (resp. 3301st) generation for the simple network (resp. the OCT network). As a result, the optimal solution is convergent in advance as smaller networks. Moreover, the proposed algorithm is validated to be suitable for the above computer networks. Please refer to Table 2 given in Appendix.

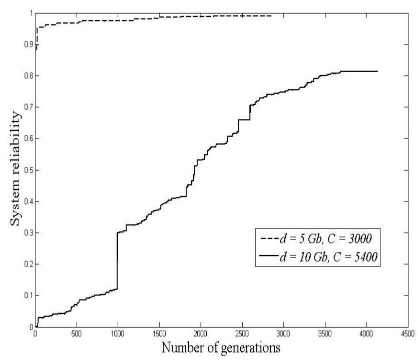

Figure 5: The progress of searching for the maximal system reliability for Figure 2.

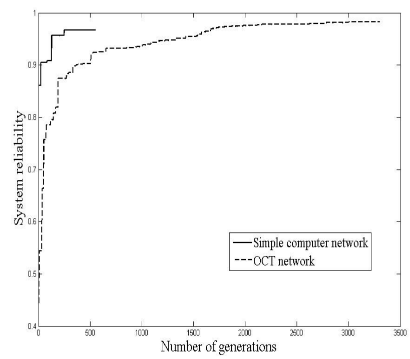

Figure 6: The progress of searching for the maximal system reliability for Figures 3 and 4.

6. Conclusions

The discussed problem in this paper has four features:

(1) There are two sets of components that are ready to be assigned to the network. One set can only be assigned to the edges and the other one can only assigned to the nodes.
(2) Each component owns several capacities due to complete failure, partial failure, maintenance, etc.
(3) Each component owns a cost per unit of data transmission.
(4) The objective of such an assignment problem is to maximize the system reliability.

We propose an efficient algorithm based on GA to solve the SRDCAB problem, in which the fitness function evaluates the system reliability of a two-type component assignment in terms of MPs and RSDP. For the network with 68 edges, 57 nodes and 17 MPs and the given 100 Type 1 components and 70 Type 2 components, the experimental results show that the optimal solution is found in no more than 86 hrs (resp. 124 hrs) as d = 5 Gb and C = 3000 (resp. d = 10 Gb and C = 5400) for the rand large network, and the optimal solution is convergent in advance as smaller d. For the simple network (resp. the OCT network), the optimal solution is found in no more than 2 (resp. 496) seconds, and the optimal solution is convergent in advance as smaller networks. Therefore, the SRDCAB-GA can be executed in reasonable time.

Acknowledgment: We would like to thank the anonymous referees for helpful and constructive comments. This work was supported in part by the National Science Council, Taiwan, Republic of China, under Grant No. NSC 96-2628-E-011-116-MY3.

References

[1] Pentico, D.W. *Assignment problems, a golden anniversary survey*. European Journal of Operational Research 2007; 176: 774-793.

[2] Wilson, J. M. *A genetic algorithm for the generalized assignment problem*. Journal of the Operational Research Society 1997; 48: 804-809.

[3] Aven, T. *Reliability evaluation of multistate systems with multistate components*. IEEE Transactions on Reliability 1985; 34: 473-479.

[4] Xue, J. *On multistate system analysis*. IEEE Transactions on Reliability 1985; 34: 329-337.

[5] Lin, J. S., C. C. Jane and J. Yuan. *On reliability evaluation of a capacitated-flow network in terms of minimal pathsets*. Network 1995; 25: 131-138.

[6] Jane, C. C., J. S. Lin and J. Yuan. *Reliability evaluation of a limited-flow network in terms of minimal cutsets*. IEEE Transactions on Reliability 1993; 42: 354-361.

[7] Lin, Y. K. *A simple algorithm for reliability evaluation of a stochastic-flow network with node failure*. Computer and Operations Research 2001; 28: 1277-1285.

[8] Lin, Y. K. *Using minimal cuts to evaluate the system reliability of a stochastic-flow network with failures at nodes and arcs*. Reliability Engineering and System Safety 2002; 75: 41-46.

[9] Lin, Y. K. *Reliability of a stochastic-flow network with unreliable branches & nodes under budget constraints*. IEEE Transactions on Reliability 2004; 53: 381-387.

[10] Lin, Y. K. *Reliability evaluation for an information network with node failure under cost constraint*. IEEE Transactions on Systems, Man and Cybernetics – Part A, Systems and Humans 2007; 37: 180-188.

[11] Lin, Y. K. *On a multicommodity stochastic-flow network with unreliable nodes subject to budget constraint*. European Journal of Operational Research 2007; 176: 347-360.

[12] Lin, Y. K. *Optimal routing policy of a stochastic-flow network*. Computer and Industrial Engineering 2009; 56: 1414-1418.

[13] Yeh, W. C. *Multistate network reliability evaluation under the maintenance cost constraints*. International Journal of Production Economics 2004; 88: 73-83.

[14] Yeh, W. C. *A new approach to evaluating reliability of multistate networks under the cost constraint*. Omega 2005; 33: 203-209.

[15] Levitin, G. and A. Lisnianski. *A new approach to solving problems of multi-state system reliability optimization*. Quality Reliability Engineering International 2001; 17: 93-104.

[16] Liu, Q., H. Zhang, X. Ma and Q. Zhao. *Genetic algorithm-based study on flow allocation in a multicommodity stochastic-flow network with unreliable nodes*. In Proceeding of The Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing 2007; pp. 576-581.

[17] Xu, W., S. He, R. Song and J. Li. *Reliability based assignment in stochastic-flow freight network*. Applied Mathematics and Computation 2009; 211: 85–94.

[18] Hsieh, C. C. and Y. T. Chen. *Reliable and economic resource allocation in an unreliable flow network*. Computer and Operations Research 2005; 32: 613-628.

[19] Hsieh, C. C. and Y. T. Chen. *Resource allocation decisions under various demands and cost requirements in an unreliable flow network*. Computer and Operations Research 2005; 32: 2771-2784.

[20] Hsieh, C. C. and M. H. Lin. *Reliability-oriented multi-resource allocation in a stochastic-flow network*. Reliability Engineering and System Safety 2003; 81: 155-161.

[21] Ramirez-Marquez, J. E. and C. M. Rocco S. *Stochastic network interdiction optimization via capacitated network reliability modeling and probabilistic solution discovery*. Reliability Engineering and System Safety 2009; 94: 913-921.

[22] Lin, Y. K. and C. T. Yeh. *Evaluation of optimal network reliability under components-assignments subject to a transmission budget*. IEEE Transactions on Reliability 2010; 59: 539-550.

[23] Colbourn, C. J. *The Combinatorics of network Reliability*. Oxford University, New York; 1987.

[24] Fisher, M. L., R. Jaikumar and L. Van Wassenhove. *A multiplier adjustment method for the generalised assignment problem*. Management Science 1986; 32: 1095-1103.
[25] Chu, P. C. and J. E. Beasley. *A genetic algorithm for the generalized assignment problem*. Computers and Operations Research 1997; 24: 17-23.
[26] Wilson, J. M. *A genetic algorithm for the generalized assignment problem*. Journal of the Operational Research Society 48 (1997) 804-809.
[27] Wang, Y. Z. *An application of genetic algorithm methods for teacher assignment problems*. Expert Systems with Applications 2002; 22: 295-302.
[28] Harper, P. R., V. De Senna, I. T. Vieira and A. K. Shahani. *A genetic algorithm for the project assignment problem*. Computers and Operations Research 2005; 32: 1255-1265.
[29] Ford, L. R. and D. R. *Fulkerson. Flows in networks*. Princeton University, New Jersey; 1962.
[30] Zuo, M. J., Z. Tian and H. Z. Huang. *An efficient method for reliability evaluation of multistate networks given all minimal path vectors*. IIE Transactions 2007; 39: 811-817.
[31] Amiri A. and H. Pirkul. *Routing and capacity assignment in backbone communication networks*. Computers and Operations Research 1997; 24: 275-87.

Yi-Kuei Lin is currently a Chair Professor of the Industrial Management Department at National Taiwan University of Science and Technology, Taiwan, Republic of China. He received a Bachelor degree from the Applied Mathematics Department at National Chiao Tung University, Taiwan. He obtained his Master degree, and Ph.D. degree in the Department of Industrial Engineering and Engineering Management at National Tsing Hua University, Taiwan, Republic of China. His research interest includes performance evaluation, stochastic network reliability, operations research, and telecommunication management.

Cheng-Ta Yeh received the B.B.A degree in Information Management from Shih Hsih University, Taiwan, the M.B.A. degree in Commerce Automation and Management from National Taipei University of Technology, Taiwan, and the Ph.D. degree in the Department of Industrial Management, National Taiwan University of Science and Technology. His research interests are mainly related to Genetic Algorithm, Network Analysis and Data Mining.

Appendix

Table 1(a): Probability distribution of capacities for 100 Type 1 components

		Capacity (Gbps)			
#	Cost (NTD)	0	1	2	3
1	14	0.000225	0.02955	0.970225	0[a]
2	3	0.095	0.905	0	0
3	7	0.005776	0.140448	0.853776	0
4	15	0.000625	0.04875	0.950625	0
5	8	0.000729	0.022113	0.223587	0.753571
6	7	0.001	0.027	0.243	0.729
7	9	0.000512	0.017664	0.203136	0.778688
8	6	0.004225	0.12155	0.874225	0
9	20	0.0004	0.0392	0.9604	0
10	10	0.000512	0.017664	0.203136	0.778688
11	13	0.000343	0.013671	0.181629	0.804357
12	16	0.015	0.985	0	0
13	14	0.0016	0.0768	0.9216	0
14	27	0.000001	0.000297	0.029403	0.970299
15	12	0.02	0.98	0	0
16	7	0.007225	0.15555	0.837225	0
17	7	0.005929	0.142142	0.851929	0

18	16	0.003	0.997	0	0
19	11	0.034	0.966	0	0
20	8	0.0036	0.1128	0.8836	0
21	22	0.000001	0.000297	0.029403	0.970299
22	13	0.000784	0.054432	0.944784	0
23	17	0.000289	0.033422	0.966289	0
24	10	0.000571787	0.018951639	0.209381361	0.771095213
25	8	0.005041	0.131918	0.863041	0
26	7	0.007396	0.157208	0.835396	0
27	2	0.067	0.933	0	0
28	12	0.001024	0.061952	0.937024	0
29	13	0.000676	0.050648	0.948676	0
30	7	0.007921	0.162158	0.829921	0
31	5	0.000512	0.017664	0.203136	0.778688
32	4	0.001	0.027	0.243	0.729
33	8	0.083	0.917	0	0
34	21	0.000015625	0.001828125	0.071296875	0.926859375
35	12	0.000274625	0.011851125	0.170473875	0.817400375
36	17	0.001369	0.071262	0.927369	0
37	25	0.000001	0.000297	0.029403	0.970299
38	10	0.000512	0.017664	0.203136	0.778688
39	8	0.006084	0.143832	0.850084	0
40	2	0.097	0.903	0	0
41	27	0.000001	0.000297	0.029403	0.970299
42	14	0.022	0.978	0	0
43	12	0.000343	0.013671	0.181629	0.804357
44	7	0.001	0.027	0.243	0.729
45	17	0.0009	0.0582	0.9409	0
46	12	0.002809	0.100382	0.896809	0
47	14	0.000166375	0.008575875	0.147349125	0.843908625
48	16	0.000125	0.007125	0.135375	0.857375
49	8	0.000857375	0.024502875	0.233422125	0.741217625
50	2	0.025	0.975	0	0
51	12	0.024	0.976	0	0
52	9	0.004096	0.119808	0.876096	0
53	10	0.003481	0.111038	0.885481	0
54	12	0.035	0.965	0	0
55	14	0.000125	0.007125	0.135375	0.857375
56	11	0.000216	0.010152	0.159048	0.830584
57	15	0.000125	0.007125	0.135375	0.857375
58	17	0.000110592	0.006580224	0.130507776	0.862801408
59	20	0.0001	0.0198	0.9801	0
60	12	0.001849	0.082302	0.915849	0
61	30	0.000000001	2.997E-06	0.002994003	0.997002999
62	21	0.000144	0.023712	0.976144	0
63	17	0.001369	0.071262	0.927369	0
64	14	0.016	0.984	0	0
65	12	0.039	0.961	0	0
66	14	0.022	0.978	0	0
67	17	0.000166375	0.008575875	0.147349125	0.843908625
68	19	0.000042875	0.003546375	0.097778625	0.898632125
69	20	0.000024389	0.002449833	0.082027167	0.915498611
70	17	0.000324	0.035352	0.964324	0
71	29	0.000000343	0.000145971	0.020707029	0.979146657
72	6	0.004356	0.123288	0.872356	0
73	3	0.055	0.945	0	0
74	11	0.001936	0.084128	0.913936	0
75	17	0.000035937	0.003159189	0.092573811	0.904231063
76	13	0.000484	0.043032	0.956484	0
77	18	0.000121	0.021758	0.978121	0
78	19	0.000256	0.031488	0.968256	0
79	14	0.001225	0.06755	0.931225	0
80	14	0.025	0.975	0	0
81	13	0.000274625	0.011851125	0.170473875	0.817400375
82	18	0.000529	0.044942	0.954529	0

83	19	0.000144	0.023712	0.976144	0
84	14	0.000216	0.010152	0.159048	0.830584
85	12	0.000117649	0.006850053	0.132946947	0.860085351
86	10	0.046	0.954	0	0
87	6	0.034	0.966	0	0
88	11	0.000512	0.017664	0.203136	0.778688
89	20	0.001	0.999	0	0
90	6	0.005476	0.137048	0.857476	0
91	7	0.000343	0.013671	0.181629	0.804357
92	26	0.000005832	0.000954504	0.052073496	0.946966168
93	6	0.071	0.929	0	0
94	8	0.002601	0.096798	0.900601	0
95	12	0.001369	0.071262	0.927369	0
96	22	0.000012167	0.001550499	0.065862501	0.932574833
97	16	0.000144	0.023712	0.976144	0
98	21	0.000132651	0.007405047	0.137791953	0.854670349
99	25	0.000027	0.002619	0.084681	0.912673
100	13	0.035	0.965	0	0

[a]The component does not own this capacity

Table 1(b): Probability distribution of capacities for 70 Type 2 components

		Capacity (Gbps)					Capacity (Gbps)		
#	Cost (NTD)	0	4	8	No. s	Cost (NTD)	0	4	8
101	25	0.001225	0.06755	0.931225	136	19	0.005929	0.142142	0.851929
102	8	0.068	0.932	0	137	19	0.004761	0.128478	0.866761
103	9	0.002601	0.096798	0.900601	138	15	0.032	0.968	0
104	13	0.000169	0.025662	0.974169	139	23	0.006724	0.150552	0.842724
105	10	0.042	0.958	0	140	22	0.000225	0.02955	0.970225
106	30	0.002025	0.08595	0.912025	141	28	0.000064	0.015872	0.984064
107	14	0.039	0.961	0	142	16	0.008649	0.168702	0.822649
108	23	0.003249	0.107502	0.889249	143	17	0.005476	0.137048	0.857476
109	22	0.005184	0.133632	0.861184	144	6	0.065	0.935	0
110	14	0.042	0.958	0	145	9	0.034	0.966	0
111	19	0.009025	0.17195	0.819025	146	7	0.002809	0.100382	0.896809
112	8	0.015	0.985	0	147	5	0.005625	0.13875	0.855625
113	11	0.037	0.963	0	148	14	0.018	0.982	0
114	12	0.022	0.978	0	149	5	0.097	0.903	0
115	17	0.006889	0.152222	0.840889	150	23	0.001089	0.063822	0.935089
116	20	0.0025	0.095	0.9025	151	5	0.1	0.9	0
117	21	0.005929	0.142142	0.851929	152	9	0.0144	0.2112	0.7744
118	15	0.012	0.988	0	153	9	0.0121	0.1958	0.7921
119	20	0.000484	0.043032	0.956484	154	6	0.099	0.901	0
120	19	0.0025	0.095	0.9025	155	6	0.007921	0.162158	0.829921
121	14	0.01	0.99	0	156	14	0.000729	0.052542	0.946729
122	16	0.001764	0.080472	0.917764	157	7	0.041	0.959	0
123	10	0.021	0.979	0	158	11	0.002809	0.100382	0.896809
124	16	0.001764	0.080472	0.917764	159	9	0.01	0.18	0.81
125	14	0.041	0.959	0	160	4	0.104	0.896	0
126	28	0.0004	0.0392	0.9604	161	6	0.063	0.937	0
127	14	0.013	0.987	0	162	9	0.046	0.954	0
128	10	0.001681	0.078638	0.919681	163	6	0.072	0.928	0
129	20	0.008649	0.168702	0.822649	164	7	0.012769	0.200462	0.786769
130	19	0.0025	0.095	0.9025	165	7	0.010201	0.181598	0.808201
131	13	0.016	0.984	0	166	10	0.012	0.988	0
132	16	0.007396	0.157208	0.835396	167	10	0.01	0.99	0
133	17	0.0064	0.1472	0.8464	168	5	0.013225	0.20355	0.783225
134	13	0.04	0.96	0	169	8	0.031	0.969	0
135	25	0.0081	0.1638	0.8281	170	7	0.006084	0.143832	0.850084

Table 2 (a): The optimal two-type component assignment of Figure 1 under d = 5 Gb and C = 3000

Maximal system reliability: 0.9896641209

i of e_i	Assigned component: s of π_s	i of e_i	Assigned component: s of π_s	i of e_i	Assigned component: s of π_s	i of e_i	Assigned component: s of π_s	i of e_i	Assigned component: s of π_s	i of e_i	Assigned component: s of π_s
1	91	22	29	43	37	64	7	85	113	106	116
2	14	23	23	44	72	65	47	86	101	107	167
3	18	24	50	45	68	66	92	87	131	108	164
4	45	25	100	46	62	67	48	88	118	109	127
5	5	26	35	47	59	68	54	89	155	110	144
6	44	27	39	48	41	69	138	90	114	111	146
7	94	28	76	49	88	70	121	91	115	112	169
8	93	29	66	50	12	71	162	92	151	113	148
9	24	30	60	51	61	72	107	93	161	114	152
10	34	31	27	52	11	73	123	94	116	115	158
11	38	32	36	53	87	74	168	95	119	116	136
12	77	33	6	54	28	75	140	96	166	117	114
13	70	34	53	55	84	76	134	97	119	118	120
14	83	35	16	56	43	77	150	98	130	119	163
15	49	36	22	57	89	78	160	99	118	120	145
16	10	37	96	58	58	79	124	100	139	121	156
17	82	38	57	59	30	80	117	101	159	122	111
18	9	39	56	60	85	81	153	102	122	123	135
19	75	40	71	61	80	82	147	103	132	124	129
20	90	41	99	62	63	83	126	104	142	125	128
21	78	42	69	63	55	84	170	105	157		

Table 2 (b): The optimal two-type component assignment of Figure 1 under d = 10 Gb and C = 5400

Maximal system reliability: 0.8136644097

i of e_i	Assigned component: s of π_s	i of e_i	Assigned component: s of π_s	i of e_i	Assigned component: s of π_s	i of e_i	Assigned component: s of π_s	i of e_i	Assigned component: s of π_s	i of e_i	Assigned component: s of π_s
1	38	22	70	43	32	64	77	85	107	106	117
2	97	23	23	44	34	65	68	86	127	107	123
3	39	24	58	45	33	66	67	87	141	108	169
4	7	25	65	46	15	67	83	88	124	109	138
5	36	26	80	47	10	68	47	89	113	110	168
6	79	27	2	48	91	69	155	90	134	111	165
7	49	28	28	49	96	70	118	91	122	112	167
8	57	29	16	50	3	71	111	92	147	113	157
9	63	30	29	51	13	72	164	93	119	114	135
10	22	31	50	52	17	73	108	94	126	115	158
11	4	32	92	53	78	74	145	95	159	116	132
12	99	33	55	54	72	75	153	96	136	117	128
13	93	34	98	55	26	76	133	97	103	118	170
14	35	35	69	56	60	77	148	98	139	119	150
15	11	36	14	57	48	78	106	99	112	120	156
16	56	37	21	58	41	79	114	100	140	121	130
17	24	38	5	59	61	80	109	101	129	122	101
18	1	39	37	60	9	81	120	102	115	123	104
19	53	40	81	61	85	82	142	103	131	124	166
20	20	41	84	62	75	83	146	104	121	125	143
21	82	42	44	63	71	84	116	105	125		