Solving Terminal Allocation Problem Using Simulated Annealing Arithmetic

Zengnian Zhang[1], Xudong Ke[2]
1 Faculty of Electronic and Information Engineering
Zhejiang Wanli University
No.8, South Qian Hu Road Ningbo, Zhejiang Province
China
2 Electronic- 053 Class, Faculty of Electronic and Information Engineering
Zhejiang Wanli University
No.8, South Qian Hu Road Ningbo, Zhejiang Province
China
http://dxxy.zwu.edu.cn

Abstract: - Due to the tremendous growth in telecommunications network, a large variety of combinatorial optimization problems have aroused people's enormous interest. In those problems, the terminal allocation attracts people's attention most. In this paper, we focus on studying the capability of Simulated Annealing Arithmetic for optimizing the terminal allocation problems in communications network. They take advantage of the best characteristic of the two effective scheduling strategies Round Robin and Shortest Distance based on local information at the terminal in the communications network. The effectiveness of the Simulated Annealing Arithmetic, where some cooling strategies are used, is passed judgment by comparing system performance under different terminal allocation algorithms including Round Robin and Short Distance. Experimental results show that the proposed Simulated Annealing Arithmetic provides an optimized combinatorial solution, therefore increase the whole throughput of the communications network system.

Key-Words: - Terminal Allocation; Simulated Annealing Arithmetic; Communications Network System;

1 Introduction

A lot of research on computer communications networks has grown explosively. This is because it is widely used in all walks of life, and this trend is increasing [1] [2]. Terminal Allocation is an important problem in telecommunication networks. It can increase the whole throughput of the networks, and decrease the cost of connections between the terminal and concentrator. The target of the terminal allocation problem implies fixing the minimum cost links to construct a network between a specified set of terminals and concentrators. The sets of terminals and concentrators have determined positions and are known. The necessary condition of each terminal, and all concentrators and the cost of connection are all known and can change. For obtaining the minimized the total systemic cost, the problem can be described as each terminal assigned to a concentrator according to some principles for Performance Optimization. First of the principles is each terminal must be associated to the only one of the concentrators. The second one is the whole power of any concentrator associated must not outmatch itself.

The above Terminal Allocation Problem can be described as follows, set:

Terminals as $l_1, l_2, ...l_T$

Weights as $w_1, w_2, ...w_T$

Concentrators as $r_1, r_2, ...r_C$

Capacities $p_1, p_2, ...p_C$

w_i is the weight, or capacity requirements of terminal l_i. The weights and capacity are positive integers and w_i is smaller or equal to $\min\{p_1, p_2, ...p_C\}$ for $i = 1,2,...T$. The *T* terminals and the *C* concentrators are place on the Euclidean grid, i.e., l_i has coordinates (l_{i1}, l_{i2}) and r_j has coordinates (r_{j1}, r_{j2}).

A simple and feasible method for this problem is to allocate each terminal node to one node of the concentrators set, so as to no concentrator overstep its capacity. Put differently, a simple and feasible method for solving the terminal allocation problem is a vector

$\vec{x} = x_1 x_2 ... x_T$,

Where $x_i = j$ means that the *ith* terminus node is allocated to the concentrator *j*, and

$1 \le x_i \le C$

x_i is an integer, for $i = 1,2,...T$

In other words, whole terminus node must be allocated, and

$\sum_{i \in R_j} w_i \le p_j$

for $j = 1,2,...C$, where

$R_j = \{i \mid x_i = j\}$. It means that the capacity of one concentrator is not overstep itself, and R_j represents the terminus node is allocated to one of the concentrators *j*.

The target of this combinatorial optimization problem is:

$$Z(\vec{x}) = \sum_{i=1}^{T} \cos t_{ij} \quad (1)$$

$$\cos t_{ij} = round\sqrt{(l_{i1} - r_{j1})^2 + (l_{i2} - r_{j2})^2} \quad 1 \le i \le T \quad (2)$$

Z, the result of the aggregate of the distance between terminus node *i* and concentrator *j*, denotes the total cost of allocating every terminus node to one concentrator according to the method represented by vector *x*.

The Terminal Allocation problem is a NP-complete combinatorial optimization problem. This means that we cannot ensure to obtain the best result in a feasible amount of time. The key of this problem is a motivation for using Simulated Annealing Arithmetic to obtain approximate, rather than exact, results.

Table 1 The coordinates and weigh of the terminal nodes

Terminal node	Coordinates	Weigh
1	54,28	5
2	28,75	4
3	84,44	4
4	67,17	2
5	90,41	3
6	68,67	1
7	24,79	3
8	38,59	4
9	27,86	5
10	07,76	4

Table 2 the capacity and coordinates of the concentrator site

Concentrator	Capacity	Coordinates
1	12	19,76
2	14	50,30
3	13	23,79

Table 1 illustrate a collection of ten terminal nodes and three concentrator sites, i.e., T =10 terminal sites and C=3. Table 1 shows the weight requirement and the coordinates based on a 10×10 Euclidean grid for each terminal site. Table 2 listed the coordinates for the concentrator sites and their capacities. The cost of allocating a terminal node to a concentrator site is the Euclidean distance between them rounded to the nearest integer.

Figure 1 indicates an allocation of the first 9 terminals node which cannot be extended to the 10th terminal node. That is because each of the 3 concentrators site is not able to provide the capacity requirement of the10th terminal node.

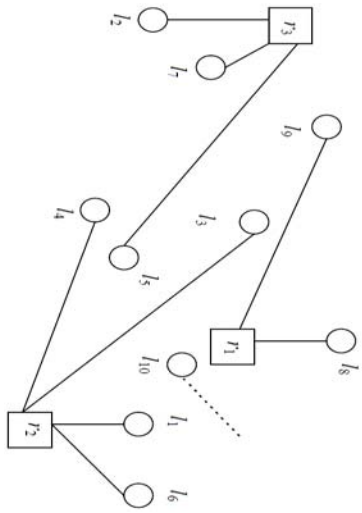


Fig. 1 Terminal nodes allocation to concentrator sites

In figure 1, the whole cost is 233 including nine terminal nodes. For inclusion the 10th terminal node into the system, we can interchange the terminal node 4th and 8th, and allocate them to concentrator 1th and 2th, the 1th concentrator have enough capacities to include 10th terminal node.

The Simulated Annealing algorithm exploits the local information of the every terminal node and uses a random number generator in an attempt to select the suitable policy; different cooling policies are analyzed to search optimal results. Our investigation was developed using Matlab as simulation tool.

The paper is structured as follows. In Section 2 we present the related works on TA problem; in Section 3 we present the algorithm we proposed; in Section 4 we describe the experimental environment; while in Section 5, we present the simulation results; in Section 6 we report about the conclusions.

2 Related Works

Some valuable efforts for the Terminal Allocation problem can be found in the literature.

For choosing the optimal design of small scale communication networks, the Simulate Annealing was proposed in [3]. Two concepts of communication network reliability were considered by authors. The first one, the 's-t' reliability, is relevant for communication between a source station and a terminal station as in the case of a two way telephone communication. The second one, the overall reliability, is a measure of simultaneous connectedness among all stations in the network. An algorithm is presented which selects the optimal set of links that maximizes the overall reliability of the network subject to a cost restriction, given the allowable node-link incidences, the link costs and the link reliabilities. The algorithm employs a variation of the simulated annealing approach coupled with a hierarchical strategy to achieve the global optimum. For complex networks, the present algorithm is advantageous over the traditional heuristic procedures. The solutions of two representative example network optimization problems are presented to illustrate the present algorithm. The potential utilization of parallel computing strategies in the present algorithm is also identified.

In [4], authors proposed Simulated Annealing Arithmetic to find solutions for packet switched communication networks. This paper presents an application of the simulated annealing heuristic to the problem of designing computer communication networks. This problem essentially consists in finding the least-cost network topologies that satisfies a given set of performance and reliability constraints.

[12] describe an implementation of the Tabu search meta heuristic that effectively finds a low-cost topology for a communications network to provide a centralized new service. Their results are compared to those of a greedy algorithm which applies corresponding decision rules, but without the guidance of the Tabu search framework. These problems are difficult computationally, representing integer programs that can involve as many as 10,000 integer variables and 2000 constraints in practical applications. The Tabu search results approach succeeded in obtaining significant improvements over the greedy approach, yielding optimal solutions to problems small enough to allow independent verification of optimality status and, more generally, yielding both absolute and percentage cost improvements that did not deteriorate with increasing problem size

A Tabu search procedure is developed to solve fibre optic communication network design problems with survivability constraints in [13]. Authors adopted Tabu Search, TS, to find a proper design of communications networks. Two systematic improving heuristics: delete-add and delete-link procedures are presented. The conditions for the candidate links to be added and deleted in the two procedures are examined by considering the feasible structures of the survivable network. A local improvement procedure is considered by combining the two heuristics for the downhill move in the search procedure. Computational results show that the proposed Tabu search outperforms the best known heuristic procedure in the literature.

A Genetic Algorithm with a penalty function as an alternative method for solving the Terminal Allocation problem was proposed in [8]. Authors proposed and compared the results with the Greedy Algorithm. The task is to allocate terminals to concentrators in such a way that each terminal is assigned to only one concentrator and the aggregate capacity of all terminals assigned to any concentrator does not overload that concentrator. Under these two hard constraints, an assignment with the lowest possible cost is sought. The proposed cost is taken to be the distance between a terminal and a concentrator.

Two different Genetic Algorithms based on Hopfield Neural Network were proposed and compared the results with the Genetic Algorithm in [9]. The proposed algorithms suit for situations in which the cost of a single assignment is not known in advance, and only the cost associated with feasible solutions can be calculated. Their approach involves a Hopfield neural network which manages the problem's constraints, whereas a GA searches for high quality solutions with the minimum possible cost. They show that our algorithm is able to achieve feasible solutions to the Terminal Allocation in instances where the cost of a single allocation in not known in advance, improving the results obtained by previous approaches. They also show the applicability of their approach to other problems related to the TA.

In [10], authors proposed a Tabu search approach to solve the problem with non-standard cost functions. A greedy decoding approach is used to generate the initial solution and then an effective and unique search approach is proposed to produce the neighbourhood, which exchange one of the terminals in each concentrator to improve the quality of solution. Simulation results with the proposed TS approach are compared with those using genetic and greedy algorithms. Computer simulations show that their approach achieves very good results in solving this problem.

[11] presents an extension of the terminal assignment problem in the case that groups of terminals must be assigned together. They analyze this situation by means of an equivalent problem: the wedding banquet problem. Authors provide a description of the problem and its mathematical definition. They also describe an application of the WBP to mobile communications network design. Two hybrid meta heuristics algorithms for the WBP are presented in the literature.

In [2], firstly, a novel chromosome representation scheme based on concentrators is proposed. This representation compares favorably against the existing terminal-based representation, which scales poorly for large problems. The results show that our evolutionary algorithms using the concentrator-based representation outperform significantly existing genetic algorithms using the terminal-based representation. Secondly, a number of new search operators used in their algorithms were also investigated empirically in order to evaluate their effectiveness for the terminal assignment problem. Finally, different combinations of evolutionary algorithms and local search are studied in this chapter. Both Lamarckian evolution and Baldwin effect have been examined in combining an evolutionary algorithm and local search. Their results show that hybrid algorithms perform better than either evolutionary algorithms or local search. However, there is no significant difference between Lamarckian-evolution-style combination and Baldwin-effect-style combination.

In the new applications of Grid computing, some significative efforts were present in [14][15][16][17]. Authors proposed a new framework for knowledge discovery based on Grid Computing. Some similar NP-Complete problem appeared in the architecture. Authors proposed some novel solution based on rough set for solving the NP-Complete problem. Rough set theory can provide us a sound solution.

3 Proposed Algorithms

In this section, we first explorer some famous Terminal Allocation algorithms, for example Round Robin algorithms and Shortest Distance algorithms applied to all Terminal nodes and concentrator sites. The communication between terminal nodes and concentrator sites conform to a wrap circle rule where each terminal node has a fixed and single neighbour to transport to, such that the direction of the packet is flowing on a circle and the length to the next node decides the quantity of packets that may be sent.

3.1 Round Robin

Round Robin is a disinterested queuing algorithm [18]. Round Robin is one of the simplest terminal allocation algorithms for processes in a network system, which allocates time slices to each terminal node in equal portions and in order, processing all terminal nodes without early or late. Round Robin algorithm is both simple and easy to carry out. Round Robin algorithm can also be implemented for other scheduling problems, such as data packet scheduling in computer networks.

After studying the Round Robin and Shortest Distance allocation algorithms, we discuss how to improve the performance using the freedom of each node to choose the best allocation strategies based on the distance information to the next node.

The selection decision is made by a Simulated Annealing arithmetic which uses the local information of the terminal nodes and concentrator sites to select the best allocation strategies, different cooling policies are analyzed to find optimal results. Our study was developed using Matlab as simulation tool.

3.2 Shortest Distance

The allocation systems apply Shortest Distance algorithm to construct connections based on the weight(x) in terminal node and concentrator site.

The allocation system running Shortest Distance picks up and transfers the data in terminals and concentrators with closest target. If the weight of queued terminal nodes in the nearest node is less than x, the concentrator search the next closest target and construct the rest connections.

3.3 Some Concept on Simulated Annealing Arithmetic

Kirkpatrick introduced Simulated Annealing in 1983 [19], which is a stochastic single agent optimization algorithm. Simulated Annealing (SA) is a generic probabilistic meta-algorithm for the global optimization problem, namely locating a good approximation to the global minimum of a given function in a large search space. It is often used when the search space is discrete (e.g., all tours that visit a given set of cities). For certain problems, simulated annealing may be more effective than exhaustive enumeration - provided that the goal is merely to find an acceptably good solution in a fixed amount of time, rather than the best possible solution.

The name and inspiration come from annealing in metallurgy, a technique involving heating and controlled cooling of a material to increase the size of its crystals and reduce their defects. The heat causes the atoms to become unstuck from their initial positions (a local minimum of the internal energy) and wander randomly through states of higher energy; the slow cooling gives them more chances of finding configurations with lower internal energy than the initial one.

By analogy with this physical process, each step of the SA algorithm replaces the Round Robin solution by a random "nearby" solution, chosen with a probability that depends on the difference between the Round Robin desponding function values and on a global parameter T (called the temperature), that is gradually decreased during the process. The dependency is such that the Round Robin solution changes almost randomly when T is large, but increasingly "downhill" as T goes to zero. The allowance for "uphill" moves saves the method from becoming stuck at local minima-which are the bane of greedier methods.

In the simulated annealing (SA) method, each point s of the search space is analogous to a state of some physical system, and the function E(s) to be minimized is analogous to the internal energy of the system in that state. The goal is to bring the system, from an arbitrary initial state, to a state with the minimum possible energy.

3.3.1 The Basic Iteration

At each step, the SA heuristic considers some neighbour s' of the Round Robin state s, and probabilistically decides between moving the system to state s' or staying in state s. The probabilities are chosen so that the system ultimately tends to move to states of lower energy. Typically this step is repeated until the system reaches a state that is good enough for the application, or until a given computation budget has been exhausted.

3.3.2 The Neighbours of a State

The neighbours of each state (the candidate moves) are specified by the user, usually in an application-specific way. For example, in the travelling salesman problem, each state is typically defined as a particular tour (a permutation of the cities to be visited); and one could define the neighbours of a tour as those tours that can be obtained from it by exchanging any pair of consecutive cities.

3.3.3 Acceptance Probabilities

The probability of making the transition from the Round Robin state s to a candidate new state *s'* is specified by an acceptance probability function *P(e,e',T)*, that depends on the energies *e = E(s)* and *e' = E(s')* of the two states, and on a global time-varying parameter *T* called the temperature.

One essential requirement for the probability function P is that it must be nonzero when *e'* > e, meaning that the system may move to the new state even when it is worse (has a higher energy) than the Round Robin one. It is this feature that prevents the method from becoming stuck in a local minimum-a state that is worse than the global minimum, yet better than any of its neighbours.

On the other hand, when *T* goes to zero, the probability *P(e,e',T)* must tend to zero if *e'* > *e*, and to a positive value if *e'* < *e*. That way, for sufficiently small values of T, the system will increasingly favour moves that go "downhill" (to lower energy values), and avoid those that go "uphill". In particular, when T becomes 0, the procedure will reduce to the greedy algorithm-which makes the move only if it goes downhill.

In the original description of SA, the probability *P(e,e',T)* was defined as 1 when *e'* < *e* - i.e., the procedure always moved downhill when it found a way to do so, Round Robin respective of the temperature. Many descriptions and implementations of SA still take this condition as part of the method's definition. However, this condition is not essential for the method to work, and one may argue that it is both counterproductive and contrary to its spirit.

The *P* function is usually chosen so that the probability of accepting a move decreases when the difference *e'* - *e* increases—that is, small uphill moves are more likely than large ones. However, this requirement is not strictly necessary, provided that the above requirements are met.

Given these properties, the evolution of the state s depends crucially on the temperature *T*. Roughly speaking, the evolution of *s* is sensitive to coarser energy variations when *T* is large, and to finer variations when *T* is small.

3.3.4 The Annealing Allocation

Another essential feature of the SA method is that the temperature is gradually reduced as the simulation proceeds. Initially, *T* is set to a high value (or infinity), and it is decreased at each step according to some annealing schedule-which may be specified by the user, but must end with *T = 0* towards the end of the allotted time budget. In this way, the system is expected to wander initially towards a broad region of the search space containing good solutions, ignoring small features of the energy function; then drift towards low-energy regions that become narrower and narrower; and finally move downhill according to the steepest descent heuristic.

3.3.5 Selecting the Parameters

In order to apply the SA method to a specific problem, one must specify the following parameters: the state space, the energy (goal) function *E()*, the candidate generator procedure *neighbour()*, the acceptance probability function *P()*, and the annealing schedule *temp()*. These choices can have a significant impact on the method's effectiveness. Unfortunately, there are no choices of these parameters that will be good for all problems, and there is no general way to find the best choices for a given problem. The following sections give some general guidelines.

3.3.6 Diameter of the Search Graph

Simulated annealing may be modelled as a random walk on a search graph, whose vertices are all possible states, and whose edges are the candidate moves. An essential requirement for the *neighbour()* function is that it must provide a sufficiently short path on this graph from the initial state to any state which may be the global optimum. (In other words, the diameter of the search graph must be small.) In the travelling salesman example above, for instance, the search space for n = 20 cities has n! = 2432902008176640000 (2.5 quintillion) states; yet the neighbour generator function that swaps two consecutive cities can get from any state (tour) to any other state in n(n − 1) / 2 = 190 steps.

3.3.7 Transition Probabilities

For each edge *(s,s')* of the search graph, one defines a transition probability, which is the probability that the SA algorithm will move to state s' when its current state is s. This probability depends on the current temperature as specified by *temp()*, by the order in which the candidate moves are generated by the *neighbour()* function, and by the acceptance probability function *P()*. (Note that the transition probability is not simply *P(e,e',T)*, because the candidates are tested serially.)

3.3.8 Acceptance Probabilities

The specification of *neighbour()*, *P()*, and *temp()* is partially redundant. In practice, it's common to use the same acceptance function *P()* for many problems, and adjust the other two functions according to the specific problem.

In the formulation of the method by Kirkpatrick et al., the acceptance probability function *P(e,e',T)* was defined as 1 if *e' < e*, and *exp((e−e')/T)* otherwise. This formula was superficially justified by analogy with the transitions of a physical system; it corresponds to the Metropolis-Hastings algorithm, in the case where the proposal distribution of Metropolis-Hastings is symmetric. However, this acceptance probability is often used for simulated annealing even when the *neighbour()* function, which is analogous to the proposal distribution in Metropolis-Hastings, is not symmetric, or not probabilistic at all. As a result, the transition probabilities of the simulated annealing algorithm do not correspond to the transitions of the analogous physical system, and the long-term distribution of states at a constant temperature *T* need not bear any resemblance to the thermodynamic equilibrium distribution over states of that physical system, at any temperature. Nevertheless, most descriptions of SA assume the original acceptance function, which is probably hard-coded in many implementations of SA.

3.3.9 Efficient Candidate Generation

When choosing the candidate generator *neighbour()*, one must consider that after a few iterations of the SA algorithm, the current state is expected to have much lower energy than a random state. Therefore, as a general rule, one should skew the generator towards candidate moves where the energy of the destination state s' is likely to be similar to that of the current state. This heuristic (which is the main principle of the Metropolis-Hastings algorithm) tends to exclude "very good" candidate moves as well as "very bad" ones; however, the latter are usually much more common than the former, so the heuristic is generally quite effective.

In the travelling salesman problem above, for example, swapping two consecutive cities in a low-energy tour is expected to have a modest effect on its energy (length); whereas swapping two arbitrary cities is far more likely to increase its length than to decrease it. Thus, the consecutive-swap neighbour generator is expected to perform better than the arbitrary-swap one, even though the latter could provide a somewhat shorter path to the optimum (with n − 1 swaps, instead of *n(n−1)/2)*.

A more precise statement of the heuristic is that one should try first candidate states s' for which *P(E(s),E(s'),T)* is large. For the "standard" acceptance function *P* above, it means that *E(s')-E(s)* is on the order of T or less. Thus, in the travelling salesman example above, one could use a *neighbour()* function that swaps two random cities, where the probability of choosing a city pair vanishes as their distance increases beyond *T*.

3.3.10 Round Robin Avoidance

When choosing the candidate generator *neighbour()* one must also try to reduce the number of "deep" local minima - states (or sets of connected states) that have much lower energy than all its neighbouring states. Such "closed catchment basins" of the energy function may trap the SA algorithm with high probability (roughly proportional to the number of states in the basin) and for a very long time.

As a rule, it is impossible to design a candidate generator that will satisfy this goal and also prioritize candidates with similar energy. On the other hand, one can often vastly improve the efficiency of SA by relatively simple changes to the generator. In the travelling salesman problem, for instance, it is not hard to exhibit two tours A, B, with nearly equal lengths, such that (0) A is optimal, (1) every sequence of city-pair swaps that converts A to B goes through tours that are much longer than both, and (2) A can be transformed into B by flipping (reversing the order of) a set of consecutive cities. In this example, A and B lie in different "deep basins" if the generator performs only random pair-swaps; but they will be in the same basin if the generator performs random segment-flips.

3.3.11 Cooling Schedule

The physical analogy that is used to justify SA assumes that the cooling rate is low enough for the probability distribution of the Round Robin state to be near thermodynamic equilibrium at all times. Unfortunately, the relaxation time-the time one must wait for the equilibrium to be restored after a change in temperature-strongly depends on the "topography" of the energy function and on the Round Robin temperature. In the SA algorithm, the relaxation time also depends on the candidate generator, in a very complicated way. Note that all these parameters are usually provided as black box functions to the SA algorithm.

Therefore, in practice the ideal cooling rate cannot be determined beforehand, and should be empirically adjusted for each problem. The variant of SA known as thermodynamic simulated annealing tries to avoid this problem by dispensing with the cooling schedule, and instead automatically adjusting the temperature at each step based on the energy difference between the two states, according to the laws of thermodynamics.

3.4 Our Proposed Simulated Annealing Arithmetic

In our proposed Simulated Annealing arithmetic, the allocation system with an initial structure with energy E is assigned in the start state at an enough large temperature T0. A random digit is also assigned and the fluctuation in energy between the start and the conversion states is measured. If change reduce internal energy, the system probability accept this new situation, otherwise the system will make a variation with some random digit from the change of temperature and energy, it can be describe formally as follows:

$$g(w) < e^{\frac{\Delta E}{T(k)}}, \text{ where } 0| < g(w) < 1 \qquad (1)$$

In the above expressions, *w* is a Gaussian random vector with probability distribution *g(w)*. After each iteration *k* the temperature *T(k)* is decreased according to a cooling condition gived and the operation is repeated until the whole minima is obtained.

The goal of this research is to interchange between Shortest Distance and Round Robin algorithms, so according to the weight between terminal nodes and concentrator site on each time piece, the quantity of the weight should be minimum. A Simulated Annealing algorithm is adopted to search the perfect switch condition so as to get the optimization results.

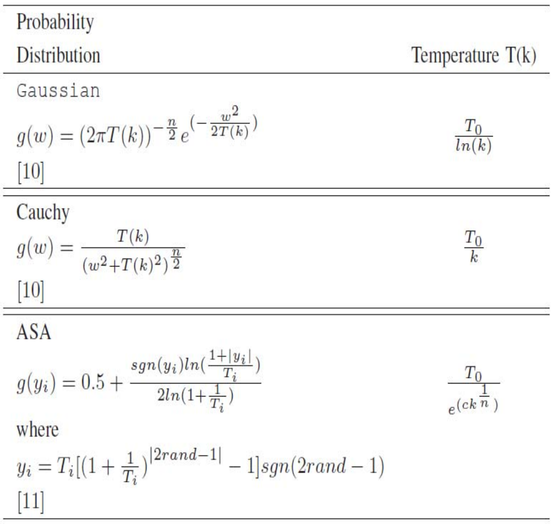
For the considered question, the assembled distance or reference distance during the real time piece and the length of the transmission represent the situation and the energy of the system in that situation respectively. The process of the computed distance value to a new one depends on the energy of the system in the actual state and in the possible new state. The mutation is always made if the latter is smaller than the former, otherwise with a probability depending on the difference between energies as well as on the present temperature of the algorithm.

The cooling schedule specifies the behaviour of the temperature and decreases it according to the time slot on the actual step of the simulation, therefore the probability of making a mutation to a worse state is higher at the beginning of the process and decreases till the point where only transitions leading to better energy reduction are allowed.

The temperature schedules investigated include the Gaussian, Cauchy and Adapted Simulated Annealing cooling disciplines whose expressions for temperature and random number generators are shown in Table 3.

According to the value of the accumulated distance as reference and the value of the actual distance between nodes, the mobile node decides to implement a discipline in such a way that SHORT DISTANCE is used if the distance between nodes is less than the accumulated distance, otherwise the ROUND ROBIN scheduling algorithm will be applied.

Table 3 Different Cooling Disciplines and Probability Distributions



Probability Distribution	Temperature T(k)
Gaussian $g(w) = (2\pi T(k))^{-\frac{n}{2}} e^{(-\frac{w^2}{2T(k)})}$ [10]	$\frac{T_0}{ln(k)}$
Cauchy $g(w) = \frac{T(k)}{(w^2+T(k)^2)^{\frac{n}{2}}}$ [10]	$\frac{T_0}{k}$
ASA $g(y_i) = 0.5 + \frac{sgn(y_i)ln(\frac{1+\lvert y_i\rvert}{T_i})}{2ln(1+\frac{1}{T_i})}$ where $y_i = T_i[(1+\frac{1}{T_i})^{\lvert 2rand-1\rvert} - 1]sgn(2rand-1)$ [11]	$\frac{T_0}{e^{(ck^{\frac{1}{n}})}}$

4 Experimental Environment

4.1 Experimental Solution

The solution includes terminal nodes waiting for connection message to other concentrator site based on wrap around method, that is, every node can send packets to its fixed next hop, so the packet flow is flowing as a circle, where every packet include a header message with target, original, and time stamp message and has a life cycle. The quantities of packets are sent to other nodes within a time piece rest with the length between nodes.

Each terminal node has a buffer array, one buffer for each possible destination, and a routing table including the index for the terminal nodes, target, length in hops to reach the target and distance of buffer that the packets are using. According to the terminal allocation algorithm packets are got to together into a send buffer using first in first out, then send it to the next node and inserted in the proper buffer. If the maximum life cycle time is arrived, the packet will be thrown off.

The memory capacity of the buffers is continued to get the action of the intelligent system trying to minimize the distance of buffer.

4.2 Model Abstraction

The problem is changed into an objective function computed as an algorithm with some nodes (M) and of hops (N) as input parameters and the aggregate length of the transmission system, the length between adjoining nodes, the routing table, medium matrix and buffer matrix of the system as outputs.

The System Allocation Algorithm describes the objective function of the considered question. It starts and influences the M Subsystem Nodes (SN) and is in charge of creating both the Buffer Matrix (BM) of all terminal nodes and concentrator sites according to the value of M and N, and the Medium Matrix (MM) that describes the medium between each pair of nodes according to the wrap around technique.

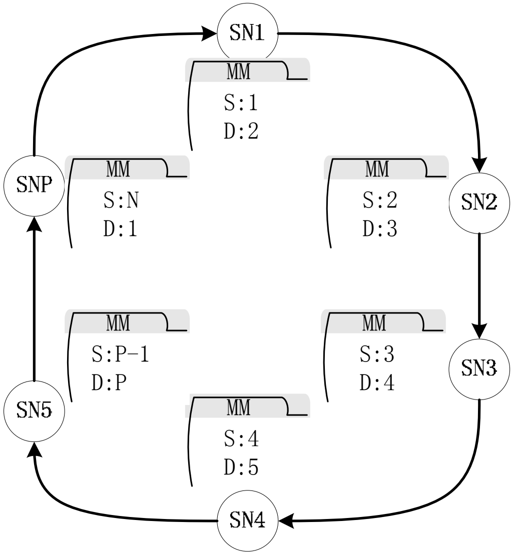


Fig. 1 System allocation Algorithm Representation.

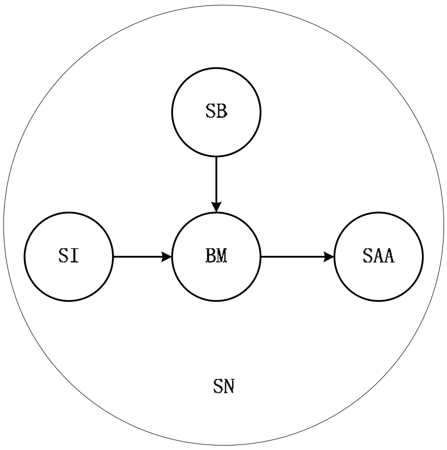


Fig. 2. Subsystem Node and its subsystem

Each terminal and concentrator nodes are represented by the Subsystem Node (Fig 2) which includes Subsystem Input, the Subsystem Buffer-SB and the Subsystem Allocation Algorithm-SAA. The

Subsystem input checks the packets addressed to the node removes them from the Medium Matrix and analyses its time stamp so as to compute the whole system capability of process and the packets thrown off. The Subsystem Buffer control the routing table and the Subsystem Allocation Algorithm apply Shortest Distance or Round Robin based on the output of the optimization algorithm.

4.3 Performance measures

For a certain experiment with fixed parameter of the situation, for example, the perimeter of the circle where the packets are sending to, the probability of packet creation and maximum number of packets that can be sent, one can get message on performance according to the outputs of the model by the following measures:

- Thrown off: the packets that cannot get their target within their life cycle time.
- Average length: Average size of every queue of a node per time piece.
- Average throughput: Average number of packets received/time piece within their lift cycle time.
- Delay: The time delay it takes to complete a packet from one node to its neighbor hop, an average value along a certain time domain is made taking into account every individual delay of each received packet.

5 Simulation Results

Average performance values are got after the experiment for four perimeters of the data flow circle (550m, 1100m, 1650m and 2200m) with 15 mobile nodes, 5 hops as maximum distance of transmission between nodes, 20% probability of packet creation and a maximum transmission of 10 packets/time piece (ts) in a time domain of 1000 ts.

First of all, we study the performance of Round Robin or Shortest Distance as allocation strategies. Figures 3a, b show that Shortest Distance has a better average throughput than Round Robin but in terms of average length of buffer, Shortest Distance supplies better results only for perimeters under 770m, this action is the major motivation for our proposal because, using the distance to the next hop as reference to decide which allocation strategy to choose, it is possible to optimize performance and minimize buffer size, see Fig 3b.

In terms of delay, see Fig. 3c, Shortest Distance shows a low and almost constant behavior compared to Round Robin and Simulated Annealing, nevertheless, the investigated optimization algorithms provide better average values than Round Robin. Also the number of thrown off, see Fig. 3d, packets is less for Shortest Distance than for the other algorithms followed by Gaussian, it is remarkable to see the high value of thrown off packets by implementing Round Robin. Computation time, see Fig 3e, is less for Short Distance and Round Robin because there is no additional processing in order to improve performance.

Analyzing the action of the different Simulated Annealing algorithms, shows to be faster than Cauchy and Gaussian as theory formulates [20]. In general terms, a better performance is obtained by carrying out Gaussian, especially the best average buffer length. For comparison reasons and in order to define whether a fair random choice between Short Distance or Round Robin would improve performance, the state is simulated in such a way that at every time piece each node randomly manages its buffers applying Short Distance or Round Robin giving 50% of probability for each option, in our case this algorithm is called”Random”.

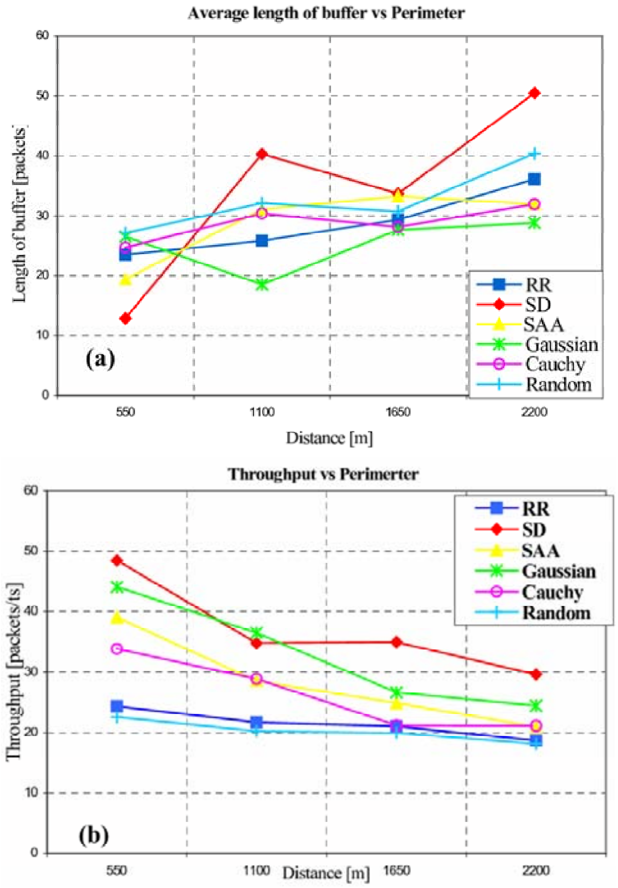
Although the”Random” selection provides a better average length of buffer than SHORT DISTANCE, the SA optimization algorithms result in better performance, as shown in figure 3.

According to the simulation results, one can establish that increasing probability of packet creation as well as distance between nodes or mobility result in an increment of the queue length.

6 Conclusion

In this paper we study the classical terminal allocation algorithms Round Robin and Shortest Distance in communication networks and the performance optimization. Based on local information of the terminal nodes, concentrator sites and a reference given by the SA optimization algorithm, each node has the possibility to choose the proper scheduling discipline according to its environment.

Simulation results show a strong relation between number of accepted mutations and performance improvement, also a balanced selection of the Round Robin or Shortest Distance allocation algorithm has a positive effect on average performance. Why the intelligent decision obtains better results compared to just a random switching between Round Robin and Shortest Distance lies in the local information of the node, specifically on its instantaneous buffer size and distance to its neighbor which, with the actual reference, are the inputs to the SA algorithms. Different SA algorithms are investigated; each of them shows advantages which may be more or less important according to the requirements of the system application. SAA is faster in terms of computation time, whereas Gaussian provides the best performance in terms of throughput and buffer size for perimeters between 700 and 1.65km. For perimeters over 1.65km Cauchy provides a better throughput than the other SA algorithms. Nevertheless, compared to SD or RR this improvement is slight. In terms of buffer size, it is strongly affected by the length of frame or perimeter and the probability of packet creation. Previous investigation results are confirmed proving the lack of influence of the buffer size on throughput, nevertheless our simulation results do not corroborate the direct proportional relation between length of buffer and delay.



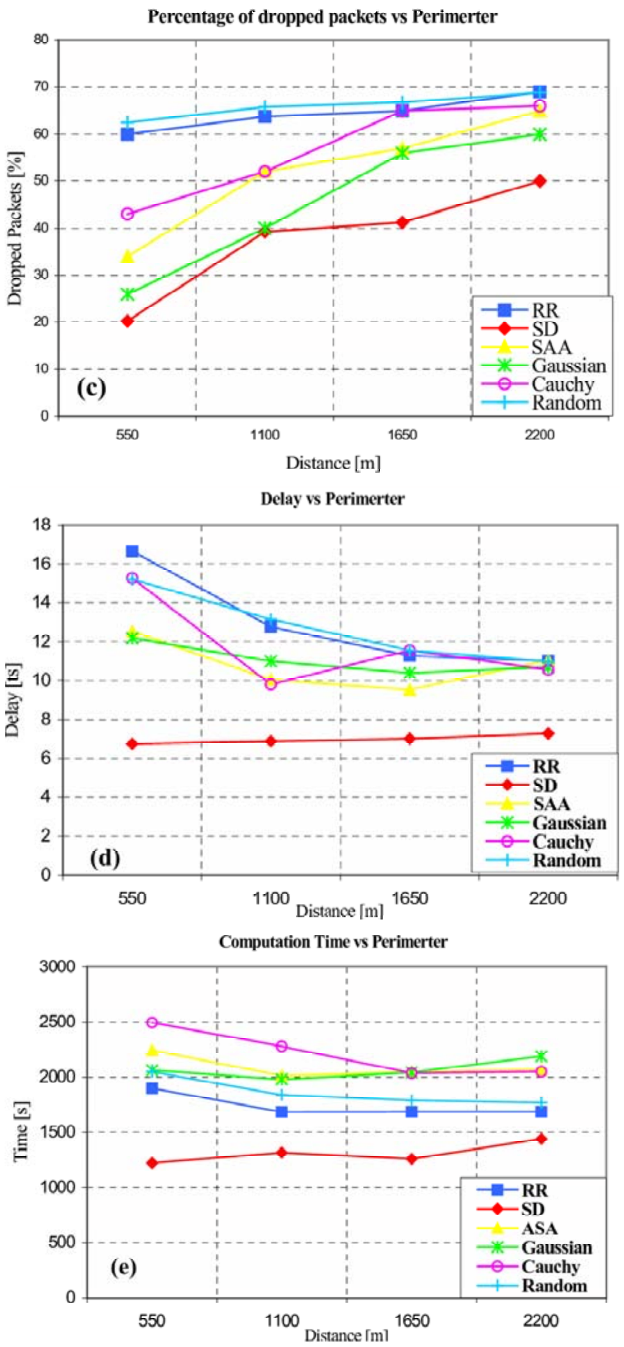


Fig. 3 Average performance values applying Round Robin and Shortest Distance Allcation algorithms and the proposed optimization algorithms

References:

[1] Salcedo-Sanz, S., Yao, X.: A Hybrid Hopfield network-genetic algorithm approach for the terminal assignment problem. In: IEEE Transaction On Systems, Man and Cybernetics, vol. 34, pp. 2343–2353. IEEE Press, USA (2004)

[2] Yao, X., Wang, F., Padmanabhan, K., Salcedo-Sanz, S.: Hybrid evolutionary approaches to terminal assignment in communications

networks. In: Hart, W.E., Krasnogor, N., Smith, J.E. (eds.) Recent Advances in Memetic Algorithms and related search technologies, pp. 129–159. Springer, New York (2004)

[3] Atiqullah, M., Rao, S.: Reliability optimization of communication networks using simulated annealing. Microelectronics and Reliability 33, 1303–1319 (1993)

[4] PieRound Robine, S., Hyppolite, M.A., Bourjolly, J.M., Dioume, O.: Topological design of computer communication networks using simulated annealing. Engineering Applications of Artificial Intelligence 8, 61–69 (1995)

[5] Glover, F., Lee, M., Ryan, J.: Least-cost network topology design for a new service: and application of a tabu search. Annals of Operations Research 33, 351–362 (1991)

[6] Koh, S.J., Lee, C.Y.: A tabu search for the survivable fiber optic communication network design. Computers and Industrial Engineering 28, 689–700 (1995)

[7] Salcedo-Sanz, S., Yao, X.: A Hybrid Hopfield network-genetic algorithm approach for the terminal assignment problem. In: IEEE Transaction On Systems, Man and Cybernetics, vol. 34, pp. 2343–2353. IEEE Press, USA (2004)

[8] Khuri, S., Chiu, T.: Heuristic Algorithms for the Terminal Assignment Problem. In: ACM Symposium on Applied Computing, pp. 247–251. ACM Press, New York (1997)

[9] Salcedo-Sanz, S., Yao, X.: A Hybrid Hopfield network genetic algorithm approach for the terminal assignment problem. In: IEEE Transaction On Systems, Man and Cybernetics, vol. 34, pp. 2343–2353. IEEE Press, USA (2004)

[10] Xu, Y., Salcedo-Sanz, S., Yao, X.: Non-standard cost terminal assignment problems using tabu search approach. In: IEEE Conference in Evolutionary Computation, vol. 2, pp. 2302–2306. IEEE Press, Portland (2004)

[11] Salcedo-Sanz, S., Portilla-Figueras, J.A., García-Vázquez, F., Jiménez-Fernández, S.: Solving terminal assignment problems with groups encoding: the wedding banquet problem. Engineering Applications of Artificial Intelligence 19, 569–578 (2006)

[12] Yao, X., Wang, F., Padmanabhan, K., Salcedo-Sanz, S.: Hybrid evolutionary approaches to terminal assignment in communications networks. In: Hart, W.E., Krasnogor, N., Smith, J.E. (eds.) Recent Advances in Memetic Algorithms and related search technologies, pp. 129–159. Springer, New York (2004)

[13] PieRound Robine, S., Hyppolite, M.A., Bourjolly, J.M., Dioume, O.: Topological design of computer communication networks using simulated annealing. Engineering Applications of Artificial Intelligence 8, 61–69 (1995)

[14] Kun Gao, Youquan Ji, Meiqun Liu, Jiaxun Chen: Rough Set Based Computation Times Estimation on Knowledge Grid. Lecture Notes in Computer Science 3470 Springer 2005, pages: 557-566

[15] Kun Gao, Kexiong Chen, Meiqun Liu, Jiaxun Chen: Rough Set Based Data Mining Tasks Scheduling on Knowledge Grid, Lecture Notes in Computer Science 3528 Springer 2005, Pages: 150-155

[16] Kun Gao: Predicting Grid Performance Based on Novel Reduct Algorithm, Lecture Notes in Computer Science 5178 Springer 2008, Pages: 289-296

[17] Kun Gao: A Uniform Parallel Optimization Method for Knowledge Discovery Grid, Lecture Notes in Computer Science 5178 Springer 2008, Pages: 306-312

[18] *Tannenbaum A,* Computer Networks. 4th edition. Prentice Hall, 307, 2003.

[19] *Kirkpatrick S.,Gelatt C.D., Vecchi M.P.,* Optimization by Simulated Annealing. Science, Volume 220, Number 4598, 13 may 1983.

[20] *Prof-Dr. Dogaru R,* Architectures and Algorithms for Natural Computing. Lecture and Seminar held during the summer term 2005 at the Technical University of Darmstadt.