International Journal of Systems Science
Vol. 40, No. 5, May 2009, 471–477

Taylor & Francis
Taylor & Francis Group

A mini–max spanning forest approach to the political districting problem

Takeo Yamada*

Department of Computer Science, The National Defense Academy, Yokosuka, Kanagawa, Japan

(*Received 23 September 2007; final version received 11 November 2008*)

We formulate the problem of political districting as a mini–max spanning forest problem, and present some local search-based heuristics to solve the problem approximately. Through numerical experiments, we evaluate the performance of the developed algorithms. We also give a case study of a prefecture in Japan for the election of the Lower House Members of the National Diet. We observe that 'hyperopic' algorithm usually gives satisfactory solutions, with the resulting districts all connected and usually balanced in size.

Keywords: political districting; heuristic algorithm; spanning forest; mini–max optimisation

1. Introduction

Gerrymandering (Morris 2006) is a form of districting in which electoral district or constituency boundaries are manipulated for an electoral advantage. To prevent this and to design more fair and reasonable districting, mathematical methods have been explored (Balinski and Young 1982) . For example, the *split-line algorithm* (Smith and Kok 2008) and the *Voronoi method* (Balinski, Brams, and Pukelsheim 2004) are based on geographical configuration of the region and divided into constituencies with artificial, piecewise linear boundaries.

Cluster analysis (Romesburg 2004) and statistical physics have been applied to this problem. These methods introduce such measures as *compactness quotients* (Nguyen and Kreinovich 1999; Bottman, Essig, and Whittle 2007) or *Hamiltonian energy* (Chou and Li 2006) to evaluate the appropriateness of the resulting districts and try to find the districting that maximises (or minimises) the sum of such measures over all constituencies. Lush, Gamez, and Kreinovich (2007) observed that naive clustering approach can lead to a disproportional representation.

In the mathematical programming approach *cost* is associated with each possible district, and the problem is usually formulated as a sort of the *set packing/covering problem* (Lawler 1976) to minimise the sum of these costs over all possible combination of districts. Some heuristic algorithms have been proposed to solve this 0-1 linear programming problem approximately, using *tabu search* (Bozkaya, Erkut, and Laporte 2003) or GRASP (Rios-Mercado and Fernandez 2009) methods. Exact algorithms have also been explored to solve this problem to optimality. These include techniques such as *column generation* and *branch-and-price* (Mehrotra, Johnson, and Nemhauser 1998), network optimisation (George, Lamar, and Wallace 1997), and capacitated transportation problem (Hojati 1996).

In all these works, the objective function was a *sum* of some measure of appropriateness over all constituencies. Unfortunately, the resulting districts can be unbalanced; we may have some very large districts together with some small ones.Also, in the optimisation approach, unless district costs are carefully defined, we may obtain disconnected districts as a part of an optimal solution.

The purpose of this article is to take connectedness and balance of constituencies explicitly into account, and present a mathematical method to solve this problem to some point of satisfaction. To this end, in Section 2 we formulate the problem as a kind of *mini–max spanning forest problem* (MMSFP), and explain the relation of this formulation to the MMSFPs studied in earlier works (Yamada, Takahashi, and Kataoka 1996, 1997). Section 3 presents two kind of heuristic algorithms to solve this problem approximately. After a numerical example for a small-sized example in Section 4, Section 5 gives a summary of numerical experiments for larger artificial instances. Finally, Section 6 describes a case study of Kanagawa Prefecture, Japan for the election of the Lower House Members of the National Diet. Through these we observe that the 'hyperopic'

*Email: yamada@nda.ac.jp

ISSN 0020–7721 print/ISSN 1464–5319 online
© 2009 Taylor & Francis
DOI: 10.1080/00207720802645246
http://www.informaworld.com

approach gives satisfactory solutions to these large-scale districting problems.

2. Political districting and the MMSFP

We model the region under consideration as an undirected *planar graph* $G=(V,E)$, where V is the set of nodes and $E \subseteq V \times V$ is the set of edges. Here each node represents an electoral unit such as cities and counties, and edges represent the adjacency relations between these units. Each node $v \in V$ has an associated integer weight $w^V(v)>0$. Usually this is the *population* of that electoral unit. Each edge $e \in E$ may also have an associated weight $w^E(e)$, which represents the *distance* or the *cost* of that edge. It is often the case that we have a set of *root nodes* $U=\{u_1,u_2,\ldots,u_r\} \subseteq V$, which implies the 'core' of each constituency. Thus, the problem is to divide the graph into connected subgraphs, each including one and only one root node, and the populations of constituencies as balanced as possible.

In graph-theoretic terms, a constituency is a connected subgraph that can be spanned by a *tree*, and a set of mutually disjoint trees that cover all nodes of G constitutes a *spanning forest* of that graph. Given a set of root nodes U, a U-rooted spanning forest F is a spanning forest of G consisting of r disjoint trees T_1, $T_2,\ldots,T_r$ such that u_i is a node of $T_i (i=1,2,\ldots,r)$. For a tree T, its weight is defined as the sum of the weights of its constituent nodes and edges. Furthermore, to obtain a balanced districting, we introduce the objective function of such a forest as

$$w(F) := \max_{1\le i\le r}\{w(T_i)\}. \qquad (1)$$

Then, the MMSFP(G,w^V,w^E,U) is to find a U-rooted spanning forest that minimises $w(F)$ over all U-rooted spanning forests of G.

In a previous paper (Yamada et al. 1996), we considered MMSFP without node weights, i.e. MMSFP$(G,\emptyset,w^E,U)$. In political districting we usually consider only populations. In this case we have MMSFP without edge weights, i.e. MMSFP$(G,w^V,\emptyset,U)$. Both of these are special cases of MMSFP(G,w^V,w^E,U), and we can convert MMSFP(G,w^V,w^E,U) into MMSFP$(G,\emptyset,w^E,U)$ using the following *node splitting* technique. That is, we transform a graph $G=(V,E)$ with node weights (and possibly with edge weights as well) into a graph without node weights. To do this, for each node $v\in V$ we prepare its *copy* v' and an edge (v,v') between these nodes. Let $\overline{V}=V\cup\{v'|v\in V\}$ and $\overline{E}=E\cup\{(v,v')|v\in V\}$, and define the graph $\overline{G}=(\overline{V},\overline{E})$. Next, we introduce edge weights to $\overline{G}$ by defining the weight for (v,v') as $w^E((v,v')) := w^V(v)$. Edge weights on E are inherited to those on $\overline{E}$.

Thus, MMSFP(G,w^V,w^E,U) can be solved by solving MMSFP$(G,\emptyset,w^E,U)$, which is denoted as MMSFP for simplicity, and hereafter we are primarily concerned with these type of problems. MMSFP with more than one root nodes is $\mathcal{NP}$-hard (Yamada et al. 1996). The problem of political districting formulated as MMSFP$(G,w^V,\emptyset,U)$ is also $\mathcal{NP}$-hard. This can be shown by direct reduction from PARTITION (Garey and Johnson 1978) to this problem for the case of complete graph K_{n+2} with integer node weights $\{0,0,w_1,\ldots,w_n\}$, where the first two are the root nodes. Indeed, the answer to PARTITION is YES if and only if we have $w(T_1)=w(T_2)=\sum_{i=1}^n w_i/2$ in MMSFP.

Previously we developed heuristic and exact algorithms to solve MMSFP with $r=2$ (Yamada et al. 1996, 1997). In this article, we discuss MMSFP with more than two root nodes, as this is usually the case in political districting. The exact algorithm (Yamada et al. 1997) is able to solve only tiny instances, so we extend our heuristic algorithm to the case of $r>2$. The algorithm is based on the *local search* (Johnson, Papadimitriou, and Yannakakis 1983) strategy, which is a sort of the *hill climbing* method (see, e.g. Papadimitriou and Steiglitz 1982). The straightforward application of this method to MMSFP is referred to as a *myopic* strategy. However, the resulting algorithm is not satisfactory in accuracy. To overcome the shortcoming of this approach, we propose a *hyperopic* strategy, and compare these strategies on a series of computational experiments. We find that the hyperopic algorithm is much superior to the myopic in solution quality.

3. Myopic and hyperopic strategies

In this section, we apply the local search method and derive two kinds of heuristic algorithms to solve MMSFP approximately. In both of the algorithms, we start from an arbitrary spanning forest and improve the forest, step-by-step, by scanning its neighbours for a better forest. If such a forest is found we take this as a new solution, and repeat the whole process until no further improvement is possible.

As for the starter, a spanning forest consisting of trees of almost balanced weights appears to be preferable. The following *greedy* method (Lawler 1976) aims to construct such a spanning forest by successively adopting the feasible edge that induces the *least* increase in the objective value of the updated forest. This algorithm usually yields a spanning forest of reasonable objective value. To describe this, let F' be a (not necessarily spanning) forest of G. For $e\in E$, $F'\cup\{e\}$ is the subgraph with e added to F'.

Let $E_{F'} := \{e \in E | e$ is incident to F', and $F' \cup \{e\}$ is a forest$\}$. Then, the algorithm is:

Algorithm GREEDY.

Step 1. Let the forest $F' = (T'_1, \ldots, T'_r)$ be initially $T'_i = (V_i, E_i)$ with $V_i := \{u_i\}$ and $E_i := \phi (i = 1, \ldots, r)$. Its value is $w(F') = 0$.

Step 2. Take $e' := \arg\min_{e \in E_{F'}} \{w(F' \cup \{e\})\}$, and put $F' \leftarrow F' \cup \{e'\}$.

Step 3. Stop if F' is a spanning forest. Otherwise, go to Step 2.

The computational complexity of GREEDY is $O(|V||E|)$, since in Step 2 it scans all the edges to find a minimising e', and Steps 2 and 3 are repeated at most $|V|$ times.

We prepare some notations before describing the procedure for improving spanning trees. Let $F = (T_1, T_2, \ldots, T_r)$ be a spanning forest of G. For an arbitrary node $u \in V$, there exists a unique path along F from this node to a root node. The corresponding root node is denoted as $r(u)$. Any nodes on the path from u to $r(u)$ are *ancestors* of u, and u is a *descendant* of such a node. By u-rooted subtree of F we mean the subgraph induced by the set of nodes u and its descendants. This is denoted as F_u. The parent node u^+ of u is the ancestor of u which is adjacent to u. An edge $(u, v) \in E$ is said to be a *bridge* of F if $r(u) \neq r(v)$.

Let us define $i^*(F) := \arg\max_{1 \leq i \leq r} \{w(T_i)\}$, which is also denoted as i^* if it is not confusing. We also introduce the set of bridges between T_i and T_j by $B(T_i, T_j) := (u, v) \in E | u \in T_i, v \in T_j\}$, where $u \in T_i$ means 'u is a node of T_i.' We further introduce $B(F) := \cup_{i \neq j} B(T_i, T_j)$ and $B^*(F) := \cup_{j \neq i^*} B(T_{i^*}, T_j)$. These represent the set of all bridges in F, and the set of those incident to the tree of maximum weight, respectively.

Given a spanning forest $F = (T_1, \ldots, T_r)$ and a bridge $e = (u, v) \in B(T_i, T_j)$, we can obtain another spanning forest $F(T_i, T_j : u, v)$ by disconnecting F_u from T_i and attaching it to T_j through e. This operation, illustrated in Figure 1, is referred to as the *swapping* of T_i and T_j with respect to (u, v). $F(T_i, T_j : u, v)$ consists of trees $T'_i := T_i \backslash F_u \backslash \{(u, u^+)\}$, $T'_j := T_j \cup F_u \cup \{e\}$ and T_l $(l \neq i, j)$. Correspondingly, the value of the forest changes from $w(F) = \max\{w(T_1), \ldots, w(T_r)\}$ to $w(F(T_i, T_j : u, v)) = \max\{w(T'_i), w(T'_j), \max_{l \neq i,j} w(T_l)\}$, where $w(T'_i) = w(T_i) - w(F_u) - w(\{u, u^+\})$ and $w(T'_j); = w(T_j) + w(F_u) + w(e)$. Let $\delta_F(T_i, T_j : u, v) := \max\{w(T_i), w(T_j) - \max\{w(T'_i), w(T'_j)\}$. This represents the degree of improvement of T_i and T_j by this swapping.

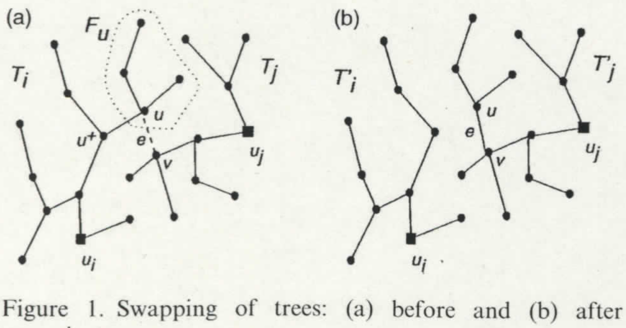

Figure 1. Swapping of trees: (a) before and (b) after swapping.

The myopic algorithm starts with the spanning forest obtained by GREEDY. Next, it tries to improve the objective value by performing a swapping with respect to some bridge. Such a bridge is taken from $B^*(F)$, since otherwise the objective value will not be improved. This process is repeated over and over again. When no further improvement is possible by such a procedure, let the spanning forest at this stage be $F = (T_1, \ldots, T_r)$, and define V_i as the set of nodes of $T_i (i = 1, \ldots, r)$. Let G_i denote the subgraph of G induced by V_i. Note that G_i $(i = 1, \ldots, r)$ are mutually disjoint. Let $\overline{T}_i$ be the minimum spanning tree within G_i. By definition $w(\overline{T}_i) \leq w(T_i)$, and thus we obtain $w(\overline{F}) \leq w(F)$ for $\overline{F} = (\overline{T}_1, \ldots, \overline{T}_r)$. Obtaining $\overline{F}$ from F is referred to as *re-optimisation* of the spanning forest F. The whole algorithm is as follows:

Algorithm MYOPIC.

Step 1. Using GREEDY, find an initial spanning forest F^0, and let $F \leftarrow F^0$.

Step 2. If there exists a bridge $(u, v) \in B^*(F)$ between T_{i^*} and some other tree T_j such that $\delta_F(T_{i^*}, T_j : u, v) > 0$, go to Step 3; else go to Step 4.

Step 3. Update the spanning forest by swapping T_{i^*} and T_j with respect to (u, v), namely, $F \leftarrow F(T_{i^*}, T_j : u, v)$. Go to Step 2.

Step 4. Re-optimise F to obtain $\overline{F}$. If $F \neq \overline{F}$, let $F \rightarrow \overline{F}$ and go to Step 2; else stop.

This algorithm is termed *myopic* since at each step it moves to a strictly improving solution. As we shall observe later in numerical experiments, the myopic strategy frequently ends up in a poor local optimum. This is because requiring a strictly improved solution at each iteration is too restrictive, and the algorithm tends to terminate at relatively early stages by failing to find a bridge as required in Step 2. Taking this into account we modify the myopic algorithm to obtain the following.

Algorithm HYPEROPIC.

Step 1. Using GREEDY, find an initial spanning forest F^0, and let $F \leftarrow F^0$.

Step 2. If there exists a bridge $(u, v) \in B(F)$ between two distinct trees T_i and T_j such that $\delta_F(T_i, T_j : u, v) > 0$, go to Step 3; else go to Step 4.

Step 3. Update the spanning forest by swapping T_i and T_j with respect to (u, v), namely, $F \leftarrow F(T_i, T_j : u, v)$. Go to Step 2.

Step 4. Re-optimise F to obtain $\overline{F}$. If $F \neq \overline{F}$, let $F \rightarrow \overline{F}$ and go to Step 2; else stop.

Except for Step 2, the algorithm is identical to MYOPIC. However, since in Step 2 we take a bridge from $B(F)$ ($\supseteqq B^*(F)$), the objective value may not be improved by such a swapping. Even if this is the case we perform swapping, hoping to obtain a better solution in later stages. Note that for $r=2$ the distinction between myopic and hyperopic algorithms vanishes since $B(F) \equiv B^*(F)$ in this case.

The computational complexity of one iteration (Steps 2 and 3) is $O(|E|)$ both in MYOPIC and HYPEROPIC, since it needs to evaluate $\delta_F(T_i, T_j : u, v)$ for all $(u, v) \in E$. Step 4 requires additional computation, but this is rarely encountered in practice.

4. An example of electoral districting

We consider electoral districting for a region represented as the graph $P_{20,46}$ of Figure 2. Here we have 20 nodes and 46 edges, and five root nodes are depicted in black. The population (in thousands) at each node is given in Table 1, and all edges are of zero weight.

Solving MMSFP for this graph we naturally obtain an electoral districting. Figure 3 depicts the results of MYOPIC and HYPEROPIC, where constituencies A to E are shown as trees. Populations of the constituencies produced by these methods are shown in Table 2, where the column of 'Unbalance' gives the ratio of the maximum population of the constituencies over the minimum.

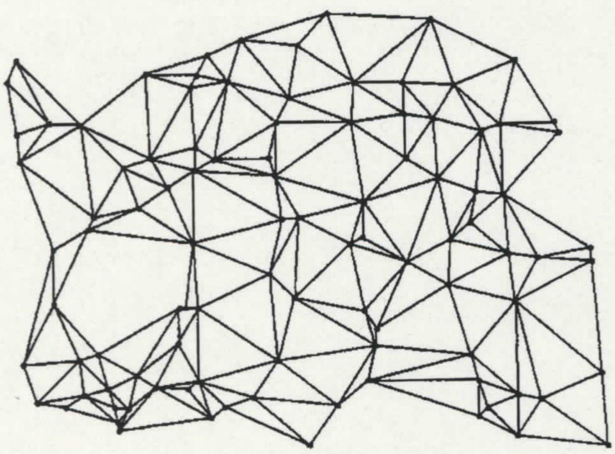

Figure 2. Planar graph $P_{20,46}$.

5. Numerical experiments

To evaluate the heuristic algorithms developed in the previous sections, especially for the districting problems of larger sizes with many root nodes, numerical experiments are carried out for planar graph $P_{n,m}$ with n nodes and m edges. We consider the case of n between 200 and 1000, and the number of root nodes is between 10 and 30. The population at each node is distributed uniformly random over the integer interval [2,20], and root nodes are randomly taken on the graph. We implemented the heuristic algorithms of Section 3 in ANSI C language and computation was done on a DELL DIMENSION 8400 computer (CPU Intel Pentium 4(R), 3.4 GHz).

Table 3 summarises the results of experiments. For each method, graph and the number of root nodes

Table 1. $P_{20,46}$ population data.

City	Population	City	Population	City	Population	City	Population
1	3	6	17	11	10	16	9
2	18	7	6	12	12	17	9
3	15	8	18	13	15	18	13
4	13	9	10	14	8	19	5
5	10	10	18	15	12	20	7

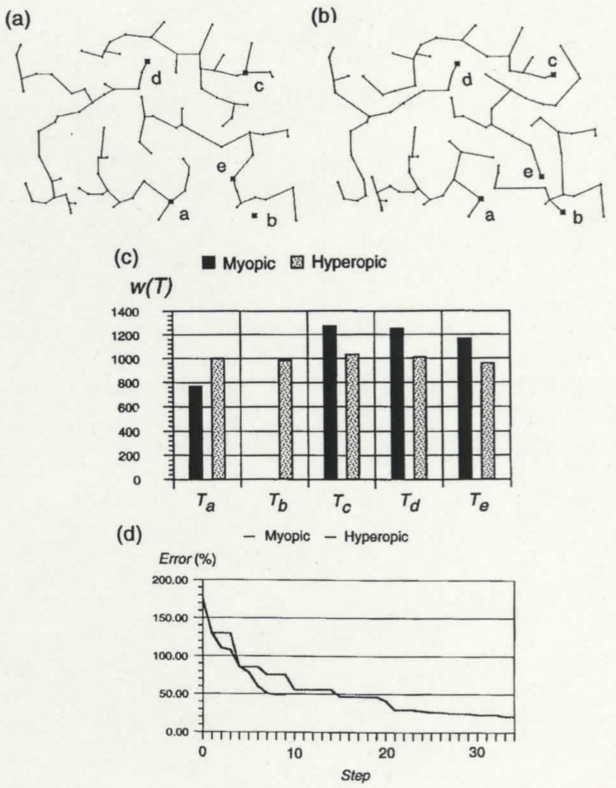

Figure 3. Electoral districting for $P_{20,46}$: (a) MYOPIC and (b) HYPEROPIC.

Table 2. Electoral districting results.

Method	Constituency A	B	C	D	E	Unbalance
MYOPIC	42	39	45	51	51	1.31
HYPEROPIC	48	47	47	42	44	1.14

Table 3. Numerical experiments.

Graph	#Roots	MYOPIC Iter.	MYOPIC Unbalance	HYPEROPIC Iter.	HYPEROPIC Unbalance
$P_{200,560}$	10	23.6	35.02	69.0	1.04
	20	31.1	31.74	115.6	1.43
	30	44.8	18.07	96.2	1.46
$P_{400,1120}$	10	58.3	13.60	95.9	1.03
	20	41.8	88.74	196.2	1.33
	30	47.6	62.14	198.6	1.60
$P_{600,1680}$	10	77.5	130.93	227.1	1.02
	20	81.6	89.34	280.3	1.37
	30	83.1	49.03	236.6	1.22
$P_{800,2240}$	10	77.5	14.07	209.6	1.01
	20	89.4	71.86	279.1	1.05
	30	100.2	77.31	348.1	1.14
$P_{1000,2800}$	10	98.6	189.31	271.4	1.02
	20	101.3	155.86	422.1	1.15
	30	125.2	123.40	405.5	1.16

(#Roots), it shows the number of iterations (Step 2–4) and the degree of unbalance (the ratio of the maximum population of constituencies over the minimum) in the columns labelled 'Iter.' and 'Unbalance', respectively. Each row is the average over 30 independent random runs. In all cases, CPU time was <1 s, and thus negligible.

For these problems, MYOPIC usually gave unsatisfactory results with unbalanced ratio frequently larger than 100. In HYPEROPIC this ratio is always <2, and often it is near to 1.0. Thus, we conclude that HYPEROPIC overperforms MYOPIC for districting problems with hundreds of electoral units and 10 or more root nodes.

6. A case study

Figure 4 is a graphical representation of Kanagawa Prefecture, Japan, where nodes and edges represent cities (including counties and wards) and their adjacency relations. Table 4 gives the population of these cities (Shimbun 1994).

Under the revised law for the election of Members of the Lower House (The Japan Times 1994), these cities are divided into 17 single-seat constituencies as shown in Figure 5(a). In this districting, the maximum and minimum populations of constituency are 585,000 and 377,000, respectively, and the ratio of these two

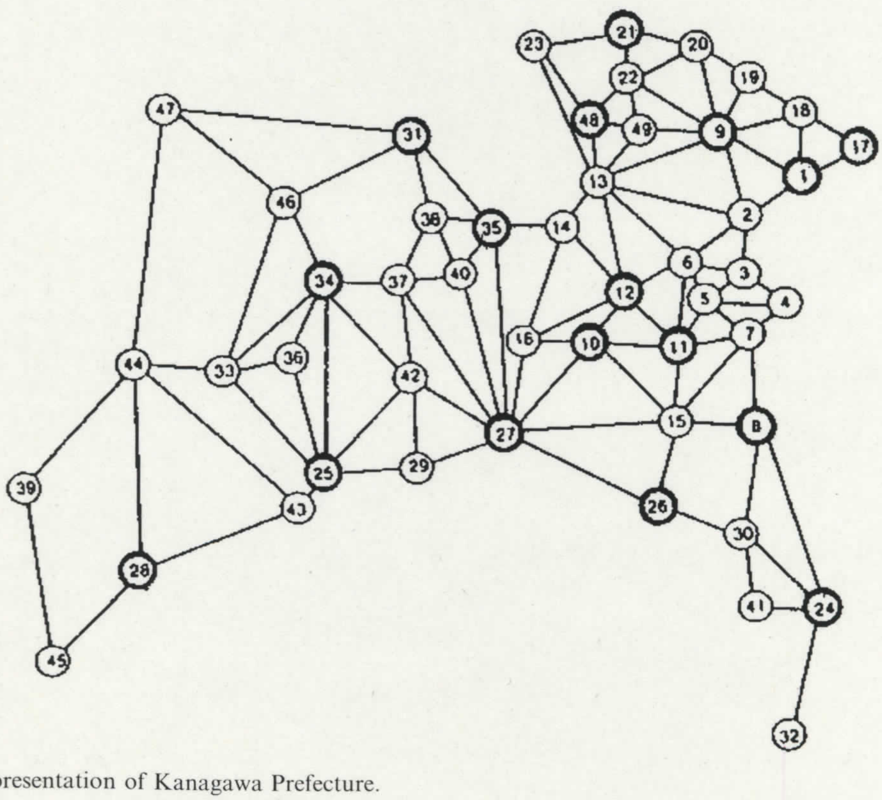

Figure 4. Graph representation of Kanagawa Prefecture.

Table 4. Population (in thousands) of the cities in Kanagawa Prefecture.

No.	City	Population	No.	City	Population	No.	City	Population
*1	Tsurumi	244.2	18	Saiwai	141.7	*35	Yamato	187.2
2	Kanagawa	200.8	19	Nakahara	181.0	36	Isehara	83.4
3	Nishi	76.2	20	Takatsu	157.8	37	Ebina	101.1
4	Naka	112.1	*21	Tama	163.7	38	Zama	107.0
5	Minami	195.8	22	Miyamae	170.2	39	M-Ashigara	42.5
6	Hodogaya	191.7	23	Aso	119.6	40	Ayase	75.9
7	Isogo	166.0	*24	Yokosuka	433.1	41	Miura-C	29.9
*8	Kanazawa	191.0	*25	Hiratsuka	238.7	42	Kouza	42.7
*9	Kohoku	293.3	*26	Kamakura	176.7	43	Naka-G	31.7
*10	Totsuka	237.0	*27	Fujisawa	341.3	44	Ashigara-K	29.5
*11	Konan	225.0	*28	Odawara	190.2	45	Ashigara-S	9.6
*12	Asahi	244.0	29	Chigasaki	197.3	46	Tsukui	14.5
13	Midori	150.0	30	Zushi	57.1	47	Aiko	13.1
14	Seya	117.3	*31	Sagamihara	508.6	*48	Aoba	14.3
15	Sakae	122.7	32	Miura	52.1	49	Tsuzuki	11.6
16	Izumi	122.2	33	Hadano	146.5			
*17	Kawasaki	193.9	*34	Atsuki	185.9			

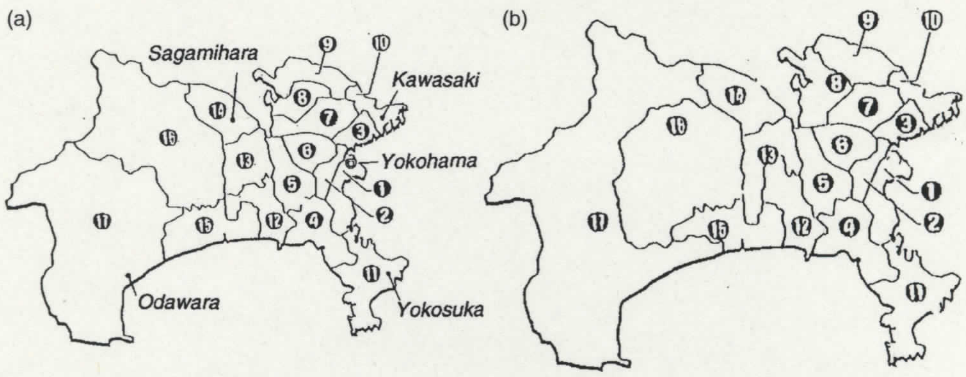

Figure 5. Districting results: (a) current and (b) HYPEROPIC.

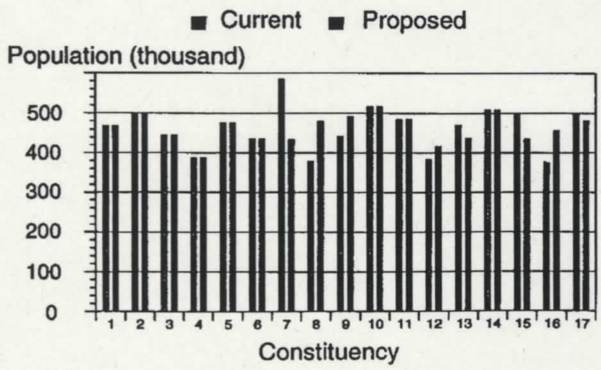

Figure 6. Population per constituency.

numbers, 155.2%, represents a measure of unbalance in the current system.

We took 17 root nodes as the biggest cities in the current constituencies, as shown with * in Table 4 and depicted in bold circles in Figure 4, and applied HYPEROPIC to this problem. The districting obtained is shown in Figure 5(b). The maximum and minimum populations of constituency in this districting is 517,000 and 388,000. In this case the unbalance ratio is 133.2%, which is 20.0% smaller than in the current system. Figure 6 illustrates the population of each constituency with the current and proposed districting.

7. Conclusions

In this article we presented two heuristics to solve the problem of electoral districting approximately. We found that HYPEROPIC gives a satisfactory solution, with the resulting districts all connected and usually better balanced in size.

However, in this work we only took population at each node into account. Edge weights, such as distance, time and cost to travel between nodes, are not explicitly considered. To obtain a more geographically acceptable districting, we need to combine these edge weights with node weights. This is an important and interesting issue for future research.

References

Balinski, M.L., Brams, S., and Pukelsheim, F. (Organizers) (7–13 March, 2004), *Analysis and Design of Electoral Systems*, Oberwolfach: Oberwolfach Meeting.

Balinski, M.L., and Young, H.P. (1982), *Fair Representation*, New Haven: Yale University Press.

Bottman, N., Essig, W., and Whittle, S. (2007), *Why weight? A Cluster-Theoretic Approach to Political Districting*, MCM2007, Seattle: Department of Mathematics University of Washington.

Bozkaya, B., Erkut, E., and Laporte, G. (2003), 'A Tabu Search Heuristic and Adaptive Memory Procedure for Political Districting', *European Journal of Operational Research*, 144, 12–26.

Chou, C.-I., and Li, S.P. (2006), 'Taming the Gerrymander–Statistical Physics Approach to Political Districting Problem', *Physica A: Statistical Mechanics and its Applications*, 369, 799–808.

Garey, M.R., and Johnson, D.S. (1978), *Computers and Intractability: A Guide to the Theory of $\mathcal{NP}$-Completeness*, San Francisco: Freeman.

George, J.A., Lamar, B.W., and Wallace, C.A. (1997), 'Political District Determination using Large-scale Network Optimisation', *Socio-Economic Planning Sciences*, 31, 11–28.

Hojati, M. (1996), 'Optimal Political Districting', *Computers & Operations Research*, 23, 1147–1161.

Johnson, D.S., Papadimitriou, C.H., and Yannakakis, M. (1983), 'How Easy is Local Search?', *Journal of Computer and System Sciences*, 37, 79–100.

Lawler, E.L. (1976), *Combinatorial Optimization–Networks and Matroids*, New York: Holt, Reinhart and Winston.

Lush, G.B., Gamez, E., and Kreinovich, V. (2007), *How to Avoid Gerrymandering: A New Algorithmic Solution*, El Paso: University Texas at El Paso. UTEP-CS-07-51a.

Mehrotra, A., Johnson, E.L., and Nemhauser, G.L. (1998), 'An Optimisation Based Heuristic for Political Districting', *Management Science*, 44, 1100–1114.

Morris, R. (November, 2006), 'The Race to Gerrymander', *The Washington Monthly*. Available online at http://findarticles.com/p/articles/mi_m1316/is_.

Nguyen, H.T., and Kreinovich, V. (1999), 'How to Divide a territory? A New Simple Differential Formalism for Optimisation of Set Functions', *International Journal of Intelligent Systems*, 14, 223–251.

Papadimitriou, C.H., and Steiglitz, K. (1982), *Combinatorial Optimization: Algorithms and Complexity*, Prentice-Hall: Englewood Cliffs.

Rios-Mercado, R.Z., and Fernandez, E. (2009), 'A Reactive GRASP for a Commercial Territory Design Problem with Multiple Balancing Requirements', *Computers & Operations Research*, 36, 755–776.

Romesburg, H.C. (2004), *Cluster Analysis for Researchers*, Morrisville, NC: Lulu Press.

Smith, W.D., and Kok, J. (2008), The Center for Range Voting, Located at http://rangevoting.org/SplitLR.html

The Kanagawa Shimbun (Newspaper) (1994), *Data Book Kanagawa 1994*, Yokohama. (in Japanese).

The Japan Times, November 22, 1994.

Yamada, T., Takahashi, H., and Kataoka, S. (1996), 'A Heuristic Algorithm for the Mini-Max Spanning Forest Problem', *European Journal of Operational Research*, 91, 565–572.

Yamada, T., Takahashi, H., and Kataoka, S. (1997), 'A Branch-and-bound Algorithm for the Mini-Max Spanning Forest Problem', *European Journal of Operational Research*, 101, 93–103.