Genetic Algorithm for Coalition formation

April, 2010 (Unfinished)

Alan Mehlenbacher

Department of Economics, University of Victoria

Victoria, B.C., Canada V8W 2Y2

Abstract

I have developed evolutionary algorithm to solve the optimal coalition structure problem and applied it to the problem of formation of criminal gangs. A previous method (Sen and Dutta, 2000) converged to a global optimum for 30 agents (with a solution space size on the order of 10^{23}), whereas this method converges to a global optimum for up to 120 agents (with a solution space size on the order of 10^{145}). A previous study of criminal gang formation (Mansour et alia, 2006) solved the problem for 3 criminals, whereas I apply the evolutionary algorithm to gang formation by 120 individual criminals and by associations of approximately 600 criminals

JEL classification: C15, C72, D83.

Keywords: coalition formation;

Author Contact:
Alan Mehlenbacher, Dept. of Economics, University of Victoria, P.O. Box 1700, STN CSC, Victoria, B.C., Canada, V8W 2Y2; e-mail: amehlen@uvic.ca; Voice: (250) 721-8537; FAX: (250) 721-6214

1 Introduction

The formation of coalitions is an active and fruitful area of investigation in international agreements and in multiagent robotic systems. There has been intensive work in developing exact and fast solutions for the multiagent systems. These algorithms typically deal with optimal coalition structures for up to 25 agents. However, we are often interested in optimal structures for 150 countries in international agreements and thousands of criminals in criminal organization structures. It therefore makes sense to turn to heuristic solutions, but this has been attempted only once previously by Sen and Dutta (2000), who borrowed an evolutionary algorithm method used in the solution of the travelling salesman problem. Their results show convergence to a global optimum for 20 and 30 agents, and convergence to a high, but not global, local optimum for 40 and 50 agents. Convergence is achieved in about 2000 generations for 20 agents and in about 4000 generations for 30 to 50 agents. Can we do better?

The generally recommended approach (Michalewicz and Fogel, DeJong) is that evolutionary algorithms should be tailored to the application. Therefore, we create new operators for the optimal coalition structure problem (OCSP) in Section 3 and show in Section 4 that they produce results that are superior to the TSP operators. We begin with a description of the OCSP in Section 2.

2 The Optimal Coalition Structure Problem (OCSP)

The term coalition applies to any group that agents agree to join. Examples include signatories to agreements (), criminal gangs(), R&D consortia (), and trade agreements (). The coalition structure is the distribution of agents in coalitions of various sizes. For example, when there are two agents, the possible coalition structures are the two agents together and the two agents separately, denoted by (12) and (1, 2). When there are three agents, the possible coalition structures are (123), (12,3), (1,23), (13, 2), and (1,2,3). The coalition structure that consists of all of the agents (in these two cases (12) and (123)), is called the grand coalition, and is usually the most desirable goal for international agreements (e.g. the Kyoto Protocol).

For solving for the optimal coalition structure, we are interested in the number of coalition structures in the solution space. As we have seen, for two agents there are two possible coalition structures and for 3 agents there are five. Table 1 shows the number of coalition structures for each n $|CS(n)|$ which is the sum of Stirling Numbers of the Second Kind $SN(n,k)$

as shown in equation (1).

$$\left|CS(n)\right| = \sum_{k=1}^{n} SN(n,k) = \frac{1}{k!}\sum_{i=0}^{k}(-1)^{k}\binom{k}{i}(k-i)^{n} \qquad (1)$$

There are two types of models that depend on the level of application of values. First, individual agents have their own values that contribute to the value of a coalition so that the coalition value is the sum of the agent values. Examples are international agreement models in which countries with their individual costs and benefits and hedonic coalition models in which individuals have preferences for other individuals. We will be using a contrived model (from Sen and Dutta, 2000) in which each agent is assigned a number n and its value is highest if it is situated in between n-1 and n+1 and if it is in a group of a certain size. In the second type of model, the coalition values depend on the size of the coalition. The example we use is for a criminal gang in which the coalition size determines some of the costs and benefits for the gang.

3 A new approach

3.1 Coalition structures and islands

Each individual coalition structure for n agents is represented by a sequence of n integers, with position i represented agent i. The integer $j \in [1,k]$ at position i represents the assignment of agent i to coalition j. The population of individuals (coalition structures) is partitioned into k islands, with each island containing individuals with j coalitions. The number of coalition structures in each island is the number of ways that the numbers 1 to n be parceled into k sets. We have n agents that we want to assign to k sets, where k goes from 1 to n and there must be at least one agent assigned to each of the k sets. We want to calculate the total number of possible assignments for each k, given n. This count is very similar to the Stirling Number of the Second Kind, and we refer to the count as a Myrvold Number: $M_{k}^{n} = \sum_{i=0}^{k-1}(-1)^{i}\,\mathrm{C}_{k}^{n}(k-1)^{n}$. The separation of the population into islands is a critical factor for success, and this concept ("parallel genetic algorithm") has been used in the TSP (Muhlenbein, 1990) and graph partitioning (von Leszewski and Muhlenbien, 1991). By isolating the islands, we maintain diversity in the population so that we do not quickly get stuck in local optima. There is interchange of diversity through a process of Migration, described in Section 3.5. As in vonLaszewski and Muhlenbein (1991), individuals are divided into islands by coalition size. Evolution and selection is done in the island. Each individual improves its fitness by local hill climbing.

Now what about the islands? If we have n agents, then we have n possible coalition sizes (1 to n), where n would be all singletons and 1 is the grand coalition. Should we have each of the CS sizes represented by a island? One possibility is to have, say, just the even CS sizes represented. On the other hand, for large n I will almost certainly have a maxsize for the number of coalitions. At least as a start, we will make the island sizes proportionate to the number of possible coalitions of size s for a given n (examples shown below). Make this number the log of the total possible coalitions.

3.2 Recombination: exchanges between individuals within an island

Individuals exchange information by a process of injection. Because of the irregular landscape, we need to maintain genetic diversity for a long time. Similar to the iterative approach used in Nagata and Kobayashi (1997), each individual "parent" in an island is paired with another island individual "mate", the mate injects a coalition into the parent, and the child replaces the parent if its value is greater. The random injection is repeated until a value improvement is achieved or until a maximum (member of the population i of size n, choose two parents randomly and then do injection until a child is found that is fitter than BOTH parents (to a max of 100 iterations). Then this child becomes member i of the new population. This maintains genetic diversity for a long period of time.

The process of injection copies whole coalitions (injects) from one parent into another. A coalition is randomly chosen in one of the parents. This is copied into the other parent. A repairing operator is applied to ensure that all coalitions are still represented.

We can vary the number of injected coalitions from 1 to up to 50% of the number of coalitions.

1	2	3	4	5	6	7	8	9	10
1	2	3	2	1	3	2	1	3	3
↓				↓			↓		
1	1	1	2	2	2	3	3	3	1
1	1	1	2	1	2	3	1	3	1

1	2	3	4	5	6	7	8	9	10
1	2	3	2	1	3	2	1	3	3
↓				↓			↓		
3	1	1	2	**3**	2	2	**3**	2	1
1	1	1	2	1	2	2	1	2	1

Migration Correction operators. Any changes to the size resulting from size recombination or mutation must be followed by an expansion or contraction operator on the membership to adjust the membership to the new size. Membership swapping never requires any correction, but a true membership crossover and mutation must preserve the completeness. Repair is required in the second example shown above. The injection of coalition 1 has erased coalition 3. We need to change some of the coalition memberships so that coalition 3 is revived. Find the lowest-value coalition and change half of the members to 3.

3.3 Self crossover

3.4 Mutation

Mutation is applied after the recombination step. To avoid creating invalid solutions, Mutation is defined as a number of exchanges of two numbers in the coding (swaps). The exchange step is repeated as long as the difference between the new descendant and one parent is under a specific limit.

A large population size and a small neighborhood size should be chosen in order to find the best results.

The local search should probably be the 2-opt heuristic of Lin and Kernighan. For all pairs of nodes, the assignment of the two nodes to a partition is exchanged, if the exchange improves the cost of the solution. This step is repeated until no improvement can be done. von Lasweski and Muhlenbein use a modification of 2-opt: Instead of trying the exchange over all pairs of nodes we execute the 2-opt algorithm only on the nodes located at the border of the partitions.

3.5 Flipping assignments

3.6 Shuffling assignments

3.7 Migration between islands.

During migration, a fixed rate of each subpopulation is selected and sent to another subpopulation. In return, the same number of migrants are received and replace individuals selected according to **some criteria**.

From Winberg and Chen: In the SBGA, the colony sends members, called migrants, back to the core. During migration the colony may send all of its members to the core or only some portion thereof. The migrants are chosen from the elite members of the colony (25% of the colony has been used). Since migration of the colony members disrupts the core group, time is given for the colony to evolve potentially useful members. The number of generations between migrations is called the migration interval. To reduce the pressure on the core even more, only a few colonies are allowed to send migrants in the same generation.

Just like all multiple-population based GAs, the SBGA needs a method to integrate the migrants arriving from the colony into the core's population. The host population is temporarily enlarged by the migrants and then pared down again to its normal size after reproduction. In other words, selection for reproduction is performed on the extended population, host members plus immigrants, but the new population will only be the size of the original population.

Consider the islands for CSsize 4 and 3. At some yet to be determined interval, we take the best CS from CSsize 4 and put it into 3. Now we have a problem because the migrant will have 4's in the CS, but it should have only threes. We want to preserve as much as possible. We can use a modification of the recombination operator. Simlarly when migrating from 3 to 4, we need to expand while keeping as many of the original 3 coalitions as possible. For this we need to calculate values for the individual coalitions in the coalition structure.

When we migrate an individual, we move it to the new island and expand or contract depending on the direction.

Expanding from n-1 to n: Calculate the values for the individual coalitions. We want to preserve the coalitions with the highest values. Therefore, take the coalition that has the smallest value and a membership of more than 1. Take half of the members of this coalition and change them from n-1 to n.

Contracting from n+1 to n: Calculate the values for the individual coalitions. We want to preserve the coalitions with the highest values. Therefore, take the members of coalition n+1 and change them to the coalition that has the smallest value.

3.8 Culling:

Now in each inner island we have two extra individuals and each outer island we have on extra individual. Sometime before the next migration, we cull the lowest value individuals.

Timing of the Operations

Evolve for 10 using recombination and mutation.

Cull the two lowest value individuals from each inner island, and the lowest value individual from the two outer islands

Migrate to n from n-1 and n+1.

(Repeat).

4.4 Sen and Dutta value

The value of a coalition structure is the sum of the values of the coalitions in the structure. The value of a coalition is the sum of the values of the individual members. For comparison with Sen and Dutta, we use their value function:

$$V_{SD} = 166.67 \sum_{S \in CS} \left(Weight(|S|) - \sum_{i \in S} (i - \min_{j} S) \right)$$

4 Application to criminal organization

4.1 Coalition structures and islands

Our evolving structure now does not need to contain individuals, but can be confined to the possible coalition sizes. We can deal with more individuals in this way. If we have n (say 1000) agents, then the possible coalition structures can be represented by coalition size strings of increasing length. The grand coalition is simply the size (1000). The coalition structures with two coalitions are represented by two size pairs: (999,1), (998,2), (997,3), (996,4), etc. Three coalitions are represented by size triplets: (998,1,1), (997, 2, 1), (996, 3, 1), (996, 2, 2), etc. This continues until we reach the maximal size coalition structure, which in this case is a $|CS|_{MAX}$ = 100-plet. In other words, there will be no more than 100 gangs when there are 1000 gangsters. In BC in 2008, with the gangster population of about 2000, there were 130 gangs. The population of potential coalition structures is segmented into $|CS|_{MAX}$ islands. For clarity and for the purposes of the Migration operator (Section 4.2), the coalition structures are always kept sorted in descending order from left to right, as shown above.

4.2 Operators

The innovation from this research is in the representation, but the operators are similar to the operators described in Section 3.. Island k has k integers representing the sizes of the k coalitions and these sizes must add up to the total number of agents n. Therefore, if after an operator changes the integers, the integers are normalized so that they sum to n. The operators for flipping, shuffling, and self-crossover are irrelevant for this coalition structure. Evolution occurs now simply by mutation, recombination, and migration.

Mutation: Say the island with three coalitions has the members coalition (910,50,40), (510,450,40), and (340,330,330). The mutation operator seeks for an improvement in value by increases of random size to a randomly chosen coalition, where the random size increase is 10% of the maximum size that is 100 in our example. For example, say it draws a random size of 25. If this were applied to coalition 1, 2, or 3 the possible mutated coalition structures are

911	901	901	515	505	505	347	337	337
50	59	50	446	455	446	327	337	327
40	40	50	40	40	50	327	327	337

Recombination: Within an island, we can have recombination in the form of classic crossover at a randomly chosen crossover point. For each member, a random mate is chosen, crossover performed at a random point, and the child replaces the member if its value is hither. This is done for each member until a

Migration: For example, say we are contracting from an island with three coalitions to one with two coalitions. For example we might be migrating the coalition (996,3,1) or (510,450,40) to an island that has members such as (999,1) or (510. 490). A reasonable method is to knock off the lowest and normalize. In this case, we would migrate (997, 3) or (531, 469). This is equivalent to distributing the third coalition size to the others in proportion to their sizes.

4.3 Criminal gang value

We now consider the problem of finding the optimal number of gangs, where a gang is a coalition and the set of gangs is the coalition structure. Consider a population of criminals of size *n* with criminals forming a coalition structure consisting of $|CS|$ gangs of size $|C_i|$. Following Mansour et alia (2006), the per unit expected punishment to gang *i* is $z_i = \alpha s|C_i|$, where α is detection effort by law enforcement and *s* is the punishment. The amount of illegal goods sold by gang *i* is q_i and $Q = \sum_{i=1}^{|CS|} q_i$. We assume that the output of a gang q_i is Cobb-Douglas in the number of members in the gang, $q_i = |C_i|^{\theta}$ so that now $Q = \sum_{i=1}^{|CS|} |C_i|^{\theta}$.

While revenue depends on the form of the demand curve, the cost for each criminal is the expected punishment which is independent of demand. The cost is a function of the length of punishment (prison time) *s* and the probability of arrest. The probability of arrest depends on the effort or investment in law enforcement α and the profile or visibility of the gang represented by the relative size of the gang $\frac{|C_i|}{\sum_{i=1}^{|CS|}|C_i|}$. The cost to an individual criminal in coalition i is $\frac{\alpha s|C_i|}{\sum_{i=1}^{|CS|}|C_i|}$.

Demand $P = \beta - \gamma Q$

Using linear demand $P = \beta - \gamma Q$ as in Mansour et alia (2006), the total revenue of a gang *i* is $R_i = (\beta - \gamma Q)|C_i|^{\theta}$. Assuming for now that the revenue is divided equally among the members of a gang, each criminal has revenue $r_i = (\beta - \gamma Q)|C_i|^{\theta-1}$. The net payoff for each gang member is therefore $\pi_i = (\beta - \gamma Q)|C_i|^{\theta-1} - \frac{\alpha s|C_i|}{\sum_{i=1}^{|CS|}|C_i|} = \left(\beta - \gamma \sum_{i=1}^{|CS|}|C_i|^{\theta}\right)|C_i|^{\theta-1} - \frac{\alpha s|C_i|}{\sum_{i=1}^{|CS|}|C_i|}$.

Demand $P = \frac{\gamma}{Q}$

Using the demand curve of Ahmed and Agiza (1998), $P = \frac{\gamma}{Q}$, the total revenue of a gang *i* is $R_i = \frac{\gamma}{Q}|C_i|^{\theta}$. Assuming for now that the revenue is divided equally among the members of a gang, each criminal has revenue $r_i = \frac{\gamma}{Q}|C_i|^{\theta-1}$. The net payoff for each gang member is therefore

$$\pi_i = \frac{\gamma}{Q}|C_i|^{\theta-1} - \frac{\alpha s|C_i|}{\sum_{i=1}^{|CS|}|C_i|} = \frac{\gamma}{\sum_{i=1}^{|CS|}|C_i|^{\theta}}|C_i|^{\theta-1} - \frac{\alpha s|C_i|}{\sum_{i=1}^{|CS|}|C_i|}.$$

Associations

Criminal associations are often based on trust, and trust in turn is based on family or extended-family relationships, previous school friendships, or previous involvement in youth gangs (CFSEU, personal communication). In this case, instead of individuals forming gangs we have associations forming gangs.

Criminal Hierarchy

An interesting extension of this approach is to assume that the revenue is not divided equally, but is divided based on the proportion of criminal types. For example, say we have an off-street distributor and a lower street-seller, with the former receiving a higher cut of the gang revenue. In this case the revenue per gang member depends on the proportion of each type in each gang. Let $|C_i^H|$ and $|C_i^L|$ denote the number of high-paid and low-paid gang members in gang *i*, and let λ be the percent premium paid to the higher-ups, then $R_i^H = \frac{(\beta - \gamma Q)\theta|C_i|}{|C_i|}$

References

De Jong, K.A., 2006. Evolutionary computation: a unified approach. MIT Press.

Watson, J., J. Watson , C. Ross , V. Eisele , J. Denton , J. Bins , C. Guerra , D. Whitley , A. Howe 1998. The Traveling Salesrep Problem, Edge Assembly Crossover, and 2-opt. Parallel Problem Solving from Nature V. Springer.

Nagata, Y., Kobayashi, S. 1997. Edge assembly crossover: a high-power genetic algorithm for the traveling salesman problem. In T. Back, editor, Proc. of the 7th International Conference on Gas, p. 450-457. Morgan Kaufmann.

Tao, G. and Michalewicz, Z. Inver-over Operator for the TSP, Proceedings of the 5th Parallel Problem Solving from Nature, T. Baeck, A.E. Eiben, M. Schoenauer, and H.-P. Schwefel (Editors), Amsterdam, September 27-30, 1998, Springer-Verlag, Lecture Notes in Computer Science, pp.803-812.

Mühlenbein, H., 1990. Evolution in time and space - the parallel genetic algorithm. Proceedings of the First Workshop on Foundations of Genetic Algorithms, Gregory J. E. Rawlins (Ed.), pp. 316-337. Morgan Kaufmann.

von Laszewski, G. and Muhlenbein, H. 1991. Partitioning a Graph with a Parallel Genetic Algorithm. Parallel Problem Solving from Nature, Lecture Notes in Computer Science, 496, pp. 165-169.

Lin, Shen; Kernighan, B. W. (1973). "An Effective Heuristic Algorithm for the Traveling-Salesman Problem". Operations Research 21: 498-516.

Michalewicz, Zbigniew and David Fogel (2002) How to Solve It: Modern Heuristics (Springer, New York)

Sen, S. and Dutta. P., (2000) Searching For Optimal Coalition Structures. IEEE Fourth International Conference On Multiagent Systems : Proceedings: 287-292

Mansour A, N. Marceau and S. Mongrain (2006), "Gangs and Crime Deterrence," Journal of Law, Economics & Organization, 22 315–339.

Tables

Table 1 Number of Coalitions Structures for Increasing n			
n	**| CS(n) |**	**n**	**| CS(n) |**
4	15	30	8.46749E+23
5	52	40	1.57451E+35
6	203	50	1.85724E+47
7	877	60	9.76939E+59
8	4,140	70	1.8075E+73
9	21,147	80	9.91268E+86
10	115,975	90	1.4158E+101
11	678,570	100	4.7585E+115
12	4,213,597	110	3.4685E+130
13	27,644,437	120	5.1263E+145
14	190,899,322	130	1.4522E+161
15	1,382,958,545	140	7.5161E+176
20	51,724,158,235,372	150	6.8206E+192

Figure 1 Results for Sen-Dutta Value Function