A Cooperative Network Game Efficiently Solved via an Ant Colony Optimization Approach

Pablo Romero[1], Franco Robledo[1] Pablo Rodríguez-Bocca[1], Darío Padula[1], and María Elisa Bertinat[1]

Facultad de Ingeniería, Universidad de la República, Herrera y Reissig 565, Uruguay[1],
promero@fing.edu.uy,
WWW home page: www.fing.edu.uy

Abstract. In this paper, a Cooperative Network Game (CNG) is introduced. In this game, all players have the same goal: display a video content in real time, with no cuts and low buffering time. Inspired in cooperation and symmetry, all players should apply the same strategy, resulting in a fair play. For each strategy we shall define a score, and the search of the best one characterizes a Combinatorial Optimization Problem (COP). In this research we show that this search can be translated into a suitable Assymmetric Traveling Salesman Problem (ATSP). An Ant Colony Optimization (ACO) approach is defined, obtaining highly competitive solutions for the CNG. Finally, we play the game in a real context, using a new strategy in a Peer-to-Peer (P2P) platform, obtaining better results than previous strategies.
Key words: COP, ATSP, ACO, P2P.

1 General Network Game

1.1 Definition

Consider a static network with $M > 1$ players and one server S who has an object, which all players are interested in. The server S cuts the object into very small pieces and, in each time slot, chooses at random only one player to send it. All players have a container that can allocate N pieces maximum, and every piece advance one position of the container in each time slot (the server always uses position 1 of the container, for only one benefited player). Each player can consult another in order to obtain a piece.

The consult works as explained next. Suppose that Player A consults Player B. Then chooses a permutation π of the natural subset $\{1, \ldots, N-1\}$, and checks if he has the piece at position $\pi(1)$. If not, checks B's container at position $\pi(1)$. If B has that piece, sends to A a copy of that piece, and the consult is successful. Otherwise, A repeats the procedure checking the container at position $\pi(2)$ and so on. Every consult finishes in a success (if the consulting player gets one piece) or in a fail (if after cheking the whole container, A does not get a piece). Players are awarded when they can fill position N of the container as many times as possible, but having at the same time the lowest number of pieces in the whole container. This will be mathematically defined next.

Each player $x \in \{1, \ldots, M\}$ chooses a permutation π_x. Be $p_x(i)$ the probability of filling position i of the container for Player x when the game is played unlimited in time (it is assumed that the number of pieces of the object is very large in relation with the container size N). Then, each player x is encouraged to choose a permutation in order to get the *continuity* $p_x(N)$ as high as possible and the expected number of pieces or *buffering time* $L = \sum_{i=1}^{N} p_x(i)$ as low as possible (see [4] for an alternative explanation in a network context).

2 Instructions for the CNG

The Cooperative Network Game (CNG) is a particular case of the General Network Game previously defined, and looks for a fair identical strategy (permutation) for all players. There is a Planner that can choose one permutation for each player. Given that he wants to have a fair and cooperative solution, he shall choose only one permutation π, which governs the consult between all players. In a stationary state, the probability of filling position i is the same for each player, named p_i. In [4] it is proved that the vector p_i complies that:

$$p_{i+1} = p_i + (1 - p_i)p_i s_i, \quad \forall i \in \{1, \ldots, N-1\} \ , \tag{1}$$

where s_i is *the strategic function* that depends on π. Specifically, for a given permutation π the strategic function s_i and the vector p_i comply the following Non-Linear System [13]:

$$(NLS(\pi)) \begin{cases} p_1 = \frac{1}{M}, \quad p_{i+1} = p_i + (1 - p_i)p_i s_i \quad \forall i \in \{1, \ldots, N-1\}; \\ s_{\pi_1} = 1 - \frac{1}{M}, \quad s_{\pi_{i+1}} = s_{\pi_i} + p_{\pi_i} - p_{\pi_i + 1} \quad \forall i \in \{1, \ldots, N-2\}. \end{cases}$$

The Non-Linear System $NLS(\pi)$ can be approximately solved for every particular π with the Newton-Raphson method, with quadratic convergence.

2.1 Score for the CNG

So far, we have not defined the objective for the CNG. There is a tradeoff between the continuity p_N and the buffering time $L = \sum_{i=1}^{N} p_i$, given that the vector p is monotonous increasing. The following analytical result will determine the score of the game:

Proposition 1. *Be X_π the random variable that counts the number of steps in a consult, needed to obtain a piece with the permutation strategy π. Then, its expected number is: $E(X_\pi) = \frac{M}{M-1} \sum_{i=1}^{N-1} \pi_i \big(p_{i+1} - p_i\big)$.*

Proof. See [13] for a detailed proof.

Thus, $E(X_\pi)$ is a linear combination of the jumps $p_{i+1} - p_i$. Moreover, it is monotonically increasing with the continuity p_N. Assuming that the whole consult is not longer than a time slot, it is convenient to maximize $E(X_\pi)$, which defines the score for the CNG:

Definition 1. *The score for the CNG for a permutation strategy π is the expected number of steps $E(X_\pi)$.*

Be P the space of all permutations of the natural subset $\{1, \ldots, N-1\}$. Consequently, in order to find the best strategy for the CNG, the next Combinatorial Optimization Problem (COP) must be solved:

$$\text{(COP)} \begin{cases} \max_{\pi \in P} E(X_\pi) \\ \text{s.t.} \quad p_i \ \textit{complies with} \quad NLS(\pi). \end{cases}$$

3 Ideal Approach for the CNG

Lemma 1. ***Imperfect Continuity:*** $p_i < 1, \forall i \in \{1, \ldots, N\}$, $\pi \in P$

Proof. We know that $p_1 = \frac{1}{M} < 1$. Suppose now that $p_h < 1$ for some $h \in \{1, \ldots, N-1\}$. Then $p_{h+1} = p_h + (1 - p_h)p_h s_h < p_h + (1 - p_h) = 1$.

Lemma 2. ***Ascendent Occupation:*** $p_i < p_{i+1}, \forall i \in \{1, \ldots, N-1\}, \quad \forall \pi \in P$.

Proof. Trivial: by induction over the set $\{1, \ldots, N-1\}$.

Lemma 3. ***Descendent Composed****: s_{π_i} is strictly decreasing with i.*

Proof. Using Lemma 2: $s_{\pi_{i+1}} = s_{\pi_i} + p_{\pi_i} - p_{\pi_i + 1} < s_{\pi_i}$.

Proposition 2. ***"Approximation Strategy Property" (ASP):***
Be $x = (x_1, \ldots, x_{N-1})$ an injective real-valued sequence. Then, there exists $\pi \in P$ that follows *the vector x in the next sense: If $x_i > x_j$ then $s_i > s_j, \quad \forall i, j \in \{1, \ldots, N-1\}$, where s is the strategic function associated with the permutation π.*

Proof. For every injective real-valued sequence x there exists a permutation of indices π such that $x_{\pi_1} > x_{\pi_2} > \ldots > x_{\pi_{N-1}}$. Using the Desendent Composed's Lemma, the strategic function s complies that $s_{\pi_1} > s_{\pi_2} > \ldots > s_{\pi_{N-1}}$. The result follows comparing the two previous inequalities.

The ASP permits to design a strategy whose occupation of the container is similar to a desired one. The ideal strategic function for a desired vector p can easily be obtained with (1):

$$s_{ideal_i} = \frac{p_{i+1} - p_i}{(1 - p_i)p_i}, \forall i \in \{1, \ldots, N-1\} \ . \tag{2}$$

We can approximate s_{ideal} via the ASP. The Follower System is illustrated in Fig. 1. Some ideal inputs were introduced into the Follower System. Here, we summarize the main experience (it is suggested to see [6]). This ideal approach *does not* reflect the behavior of chosen inputs. Rather, it is like a *dirty mirror*. However, the experience gained with chosen inputs permits to outstand a particular Subfamily of strategies defined next.

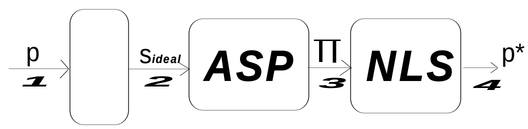

Fig. 1. Follower System: receives a desired probability of occupation p and returns a feasible occupation p^*, close to that occupation of the input. It permits to obtain the strategy π in Step 3, that achieves p^*.

Definition 2. *Subfamily of Strategies $Sub(I, J)$*
For each pair of naturals $(I, J) : I + J < N$, we will call $Sub(I, J)$ to the subfamily of strategies that can be expressed in the next way:

$$\pi(i) = N - i, \quad i = 1, \ldots, I, \quad \pi(I + j) = j, \quad j = 1, \ldots, J \ ;$$

$$\pi(I + J + k) = \left\lfloor \frac{N + J - I}{2} \right\rfloor + \left\lceil \frac{k}{2} \right\rceil (-1)^{k+1}, \quad k = 1, \ldots, N - I - J - 1 \ .$$

4 Feasible Approach based on Ant Workers

From now on, we will attend the CNG trying to find the best strategy, on the lights of the score defined in the COP of Subsection 2.1. Considering high containers (e.g. $N > 15$) an exhaustive search for the best permutation results computationally prohibitive. In this section we will translate the COP into a suitable Assymmetric Traveling Salesman Problem (ATSP). This last is solved heuristically following an Ant Colony Optimization (ACO) approach, which is inspired in the way ants find the shortest path between their nests and their food [10]. The reader can find a deep analysis of this nature-inspired metaheuristic in [11, 1–3].

Proposition 3. *"Translation of the Problem": An $N-$clique permits to obtain a bijection between a directed cycle that visits all its nodes and a permutation of $\{1, \ldots, N - 1\}$.*

Proof. Take any directed cycle $C = \{v_N, v_1, v_2, \ldots, v_{N-1}, v_N\}$.
Then, $\pi(i) = v_i$, $i = 1, \ldots, N - 1$ is clearly bijective.

Definition 3. *The function $d(\pi, \pi^*)$ (where π and π^* are permutations) counts the minimum number of swaps of elements to transform π into π^*.*

Proposition 4. *(P, d) is a metric space, being P the space of permutations.*

A Local Search can be defined substituting a permutation π with its best neighbor (which is at distance 1 from π). All these tools are used to define an ACO-based Algorithm, which finds high competitive strategies for the CNG:

```
Main Algorithm                          | Function: ApplyACO
1: d(E)   = Edges(ants)                 |1: Quality = Greedy
2: τ(E)  = Pheromones(Sub(I,J))         |2: FOR i = 1 TO ants
3: π      = ApplyACO(d,τ,iter,α,β,ρ)    |3:  π(i) = CycleACO(τ,Distances,α,β)
4: π_out = LocalSearch(π)               |4:  τ    = NewPheromones(ρ,τ,Q,Q_max)
5: RETURN π_out                         |5: RETURN MostVisitedCycle(π(1),..,π(ants))
----------------------------------------------------------------------------------------
Function: Edges                           | Function: Pheromones
1: Distances = 1                          | 1: τ = 1
2: Quality   = Greedy                     | 2: Quality = Greedy
3: FOR i = 1 TO ants DO                   | 3: FOR EACH π  ∈  Sub(I,J)
4:   π(i)     = VisitCycle(Distances)     | 4:   Q = Quality(π)
5:   Distances = UpdateCost(π(1),...π(i)) | 5:   τ = UpdateCost(π(1),..,π)
----------------------------------------------------------------------------------------
```

In the first stage of the Main Algorithm (Line 1), a non-negative assymmetric cost for each edge is initialized, with a learning mechanism based on ant exploration. The second block (Line 2) prepares the ACO application, via an initialization of the pheromones, which will permit to track cycles with high quality. The third block (Line 3) is the ACO application itself, which returns a strategy π. Finally, a local improvement is realized considering a typical local search.
The Function *Edges* translates the COP into an ATSP. To start, the cost of all edges are initialized to 1, and the Greedy strategy ($\pi_i = N - i$) is considered as a reference score. Then, each ant chooses probabilistically the next node to visit without making cycles. In the Function $VisitCycle$, ants choose the next step according with the next probabilities:

$$p(x_{j+1}) = \frac{Distances(x_j, x_{j+1})^{-1}}{\sum_{i \in NoCycle} Distances(x_j, x_i)^{-1}} \ . \tag{3}$$

So, shorter tours are preferable. Line 5 updates $Distances$. Function $UpdateCost$ finds the best strategy so far. Then, all edges in $\pi(i)$ are updated according with its score: $Distance(edge(j) \in \pi(i)) = 10(N-j) \times \frac{Q_{max}}{Q_{\pi(i)}}$, where Q_{max} is the best score obtained so far, and $edge(j)$ is the edge visited in order j in the cycle $\pi(i)$. There is an additional factor $N - j$, that avoids revisiting a cycle many times.

The pheromones for the immediate ACO application are initialized according with the experience obtained from the subfamily $Sub(I, J)$. Function $Pheromones$ follows the same structure of $Edges$. The main difference between them is the deterministic cycles used in $Pheromones$, exploiting the high quality that provides the SubFamily $Sub(I, J)$ (see [6] for more details of the quality of $Sub(I, J)$). Function *ApplyACO* goes in parallel with a traditional ACO implementation, but each ant constructs one strategy. As a consequence, the exploration mechanism is slightly different. In Line 3, each ant makes a biased walk according with the probabilities:

$$p(x_{j+1}) = \frac{\tau(x_j, x_{j+1})^{\alpha} Distances(x_j, x_{j+1})^{-\beta}}{\sum_{i \in NoCycle} \tau(x_j, i)^{\alpha} Distances(x_j, i)^{-\beta}} \ , \tag{4}$$

where x_j is step j of the cycle, and α and β are the classical priority to pheromones and costs respectively. The updating of pheromones runs similar to classical implementations, based on an evaporation factor $\rho : 0 \leq \rho \leq 1$:

$$\tau(x_j, x_{j+1}) = (1-\rho)\tau(x_j, x_{j+1}) + \rho \times \frac{10(N-j)Q}{Q_{max}} \ , \tag{5}$$

where the factor $10(N-j)$ provides a similarity of magnitudes between pheromones and distances. $LocalSearch$ chooses the best permutation among those which are at distance one from the output permutation of Function $ApplyACO$.

Theorem 1. *Be N the buffer size and $T(N)$ the average time for evaluating the quality $E(X_\pi)$. If we assume that the number of ants and the maximum number of iterations have order $O(N)$, then the average total time for running the Main Algorithm is $O(N^3 T(N))$.*

Proof. $LocalSearch$ dominates in computational effort, being C_2^{N-1} the number of neighbors for a given permutation. If the number of iterations is linear with N, the computational time for running $LocalSearch$ is $O(N^3 T(N))$.

5 Numerical Results

5.1 Comparison with Historical Strategies

There is an important reason to play the CNG: it is closely related with the delivery of live streaming contents in P2P networks. Basically, the players are now peers (computers) and the container is the storing capacity for each peer. Moreover, if the object to share is video, p_N is now the continuity of reproduction, and L is the buffering time. In P2P networks, two historical strategies for cooperation are the *Rarest First* $\pi_i = i$ and *Greedy* $\pi_i = N - i$ [4]. The former works properly in downloading, but not in streaming. The adaptation of ACO's parameters is based on [7], and then tuned in accordance with our particular problem. Our final implementation used the Main Algorithm with $\alpha = 0.4, \beta = 1.5, \rho = 0.5$ and 100 ants, for the common-network parameters $N = 40$ and $M = 100$. Fig. 3 shows that the obtained permutation achieves an excellent continuity and at the same time a latency comparable with the one reached by Greedy.

The obtained permutation was applied into a real platform named *GoalBit*, which is the first open-source P2P network that widely offers live video streaming to final Internet users [8, 9]. GoalBit mantains the BitTorrents philosophy [12] considering the tit-for-tat strategy with optimistic unchoking, extending the success in the peer selection process. The clear weakness of BitTorrent for streaming applications is its peer selection strategy: Rarest First. The analysis realized in this paper shows its unacceptable latencies.

Fig. 2. Performance of different strategies.

Strategies	Continuity	Latency
Rarest First	0.9571	21.0011
Greedy	0.9020	4.1094
Main Algorithm	0.9998	7.9821

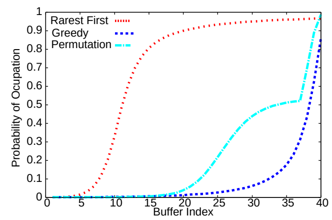

Fig. 3. Comparison between different strategies

5.2 Results

Three strategies were considered to analyze their performance: Rarest First, Greedy and the Subfamily member $Sub(16, 1)$. The parameters $I = 16$ and $J = 1$ were tuned via an exhaustive search among all strategies $\pi \in Sub(I, J)$. The test case considers $N = 40$ and 45 peers (players) entering the network. Fig. 4 shows for each strategy and each peer, the initial buffering time, number of re-bufferings and mean buffering time, respectively. The Rarest First strategy has unacceptable start-up latencies for streaming purposes. On the other hand, the new permutation presents lower start-up latencies than Greedy for most of the peers. The interruption of the video signal is clearly higher for the Greedy strategy. Rarest First trades off latency for reduced cuts, but as seen before, has latencies in the order of minutes. Only four peers experimented longer cuts when our permutation strategy was applied. However the performance of the permutation strategy was higher in the rest of the peers, with respect to classical strategies.

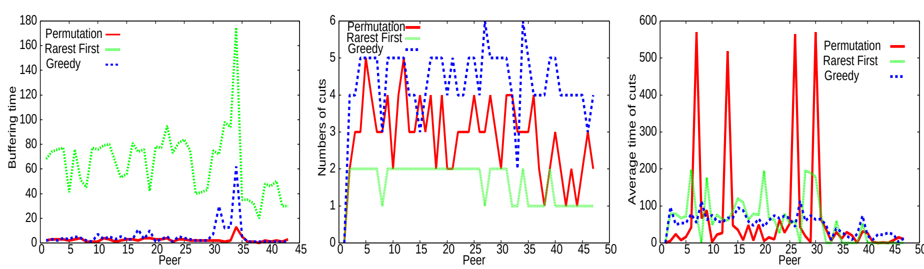

Fig. 4. Buffering time, Numbers of cuts and Average time of cuts when applying different strategies in GoalBit.

6 Conclusions

An in-depth analysis for the CNG was presented, from both an ideal and feasible approaches. Although the ideal approach fails, it gives an insight for the design of competitive strategies. Feasible strategies were found via an Ant-Worker Algorithm, named the Main Algorithm. It translates an optimization problem into an assymmetric TSP, naturally explored via ACO. A Local Search phase finally improves ACO's output. The Main Algorithm was developed and tested with common-network values of players and capacity ($M = 100$ and $N = 40$). Theoretically, it returned a strategy which achieves excellent continuity (very close to 1) and a reasonably low buffering time. Finally, the interest of playing the CNG could be verified when strategies were applied into a real P2P platform named GoalBit. A Subfamily member showed important advantages with respect to classical strategies. From both, theoretical and practical focuses, the results were highly competitive with respect to Greedy and Rarest First strategies.

References

[1] Dorigo, M., Stutzle, T.: Ant Colony Optimization. MIT Press (2004)

[2] Blum, C: Ant colony optimization: Introduction and recent trends. Physics of life Reviews, vol. 2, 353–373 (2005)

[3] Dorigo, M., Gambardella, L.: Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. IEEE Transactions on Evolutionary Computation vol. 1, 53–66 (1997)

[4] Zhou, Y., Ming, D., Chiu, Lui, J.: A Simple Model for Analyzing P2P Streaming Protocols. IEEE International Conference on Network Protocols (ICNP'07), 226–235 (2007)

[5] Oncan, T., Altynel, K., Laporte, G.: Dorigo, M., Gambardella, L.: A comparative analysis of several asymmetric traveling salesman problem formulations. Computers & Operations Research. vol. 36, n. 3, 637–654 (2009)

[6] Bertinat, M., De Vera, D., Padula, D., Robledo, F., Rodríguez-Bocca, P., Romero, P., Rubino, G.: Systematic Procedure for Improving Continuity and Latency in a P2P Streaming Network. In: Proceedings of the IEEE Latin-American Conference on Comunications (LatinCom, 2009)

[7] Duan, H., Ma, G., Liu, S.: Experimental Study of the Adjustable Parameters in Basic Ant Colony Optimization Algorithm. IEEE Congress on Evolutionary Computation, vol. 1, 149–156 (2001)

[8] Bertinat, M., De Vera, D., Padula, D., Robledo, F., Rodríguez-Bocca, P., Romero, P., Rubino, G.: GoalBit: The First Free and Open Source Peer-to-Peer Streaming Network. In: Proceedings of the of the 5th International Latin American Networking Conference. ACM Digital Library, 49–59 (LANC 2009)

[9] GoalBit Home Site, http://goalbit.sf.net

[10] R. Beckers, R., Deneubourg, J., Goss, S.: Trails and U-turns in the selection of the shortest path by the ant Lasius Niger. Journal of Theoretical Biology, vol. 159, 397–415 (1992)

[11] Dorigo, M., Birattari, M., and Stutzle, T.: Artificial Ants as a Computational Intelligence Technique. Institut de Recherches Interdisciplinaires, Universit Libre de Bruxelles, vol. 23 (2006)

[12] Cohen, B.: Incentives Build Robustness in BitTorrent. Vol. 1, 1–5, (2003).

[13] Bertinat, M., De Vera, D., Padula, D., Robledo, F., Rodríguez-Bocca, P., Romero, P., Rubino, G.: A COP for Cooperation in a Peer-to-Peer Streaming Protocol. In: International Conference in Ultra Modern Telecommunications (ICUMT 2009)