CEVOP: A Co-evolutionary Planning Methodology

Bora I. Kumova

The University of Birmingham; School of Computer Science
Birmingham; UK
B.I.Kumova@cs.bham.ac.uk; http://cs.bham.ac.uk/~bik

Abstract. The planning process of multiple autonomous agents is influenced by various factors, such as homogeneity/heterogeneity, co-operation/competition, cardinality, and communication syntax/semantics. Analytical planning approaches were usually designed for specific domains and therefore are relative domain-depend. In this work, these difficulties are addressed through a generic algorithmic approach, namely the Co-evolutionary Planning (CEVOP) methodology, which provides a planning mechanism for multiple co-operating and/or competing homogeneous and/or heterogeneous agents through a single co-evolutionary algorithm. CEVOP is rather a framework for mapping the planning problem from the multi-agent domain onto the co-evolutionary algorithm domain, in order to find concrete plans to be executed later by the agents co-operatively and/or competitively.

Keywords. Multi-agent Systems; Evolutionary Computation; Planning; Multi-agent Co-operation, Competition, Co-ordination

1. Introduction

Planning in multi-agent systems requires communication, co-operation, and control structures to be considered in the planning process as well as within a plan for its later implementation. Planning alternatives, with respect to the grade of distribution, can be summarised as follows: [9]

- Centralised planning
- Distributed planning with centralised co-ordination
- Distributed planning

An early example for distributed planning with centralised co-ordination is the global partial planning proposal [8], where the individual partial plans of the agents must be merged to construct a global one for all. The difficulty here is to define a generic algorithm for merging that is domain-independent.

A general approach for distributed planning is for instance, the co-operative problem solving approach, expressed as a theory in a quantified multi-modal logic. It consists of the four stages: [27]

- Recognition, in which an agents identifies the potential for co-operation.
- Team formation, in which the agent solicits assistance.
- Plan formation, in which the newly formed collective attempts to construct an agreed joint plan.

- Execution, in which the members of the collective play out the roles they have negotiated.

Other multi-agent planning approaches are automated planning for open architectures [24], where the adaptation service for end-to-end network connections is based on an automated planning approach; an integrated planning and scheduling approach combining sub-tasking and virtual clustering of agents with a modified Contract Net protocol [20]; the planning component of the RETSINA information gathering framework interleaves planning and execution actions to solve the problem of partial domain knowledge for the purpose of information fusion [21]; the proposal of the multi-agent planning system IMPECTing SHOP (Simple Hierarchical Ordered Planner) that interacts with external information sources [7].

Analytical approaches are solutions for specific domains and can hardly be applied in other domains. Even, after filtering out domain-specific knowledge, the algorithmic part still depends on the domain for which it was initially designed. On the other hand, in analogy to the natural adaptation processes, which are based on random but goal-driven exploration of a search space, evolutionary computation has been proven to be the most generic search approach. The utilisation of evolutionary computation for automated learning of meta rules through generalisation of primitive rules of a given domain is a promising approach for domain-independent planning.

For historical reasons, rule-based system are sometimes called classifier systems. Initial work that combined classifier systems with evolutionary computation was for the purpose of learning, but not necessarily planning. According the encoding of rules in chromosomes, two strategies were common. In the Pitt Approach, an entire rule set is represented with a chromosome, whereas, in the Michigan Approach, a single rule is represented with a chromosome [5]. An improvement of the learn efficiency of SAMUEL [10], [11] was achieved with genetic algorithms. A system that can learn and adapt reactive decision rules from simulations of multi-agent environments by using genetic algorithms [12]. GABIL continuously learns and refines concept classification rules by using a genetic algorithm [6]. Lamarckian learning in the multi-agent environment of SAMUEL [11]. Evolving reactive agents for the central food foraging problem by using genetic programming [15].

A typical classifier system is one which simulates a game. Since the domain of a game is complete, with respect to its rules, it provides an ideal environment for searching for related game strategies and allows relative easy justification of the results. Furthermore, it can be used as a benchmark. For instance, co-evolution of agents that simulate the game of Tag [25], Tic Tac Toe [26], Backgammon [23], Poker [3].

Evolutionary or co-evolutionary computation for the purpose of planning has an even shorter history and was less discussed in the literature. For instance, co-evolutionary computation for path planning [22]; the automatic programming of agents that learn mental models and create simple plans of action [1]; global planning from local eyeshot, which is a implementation of observation-base plan co-ordination in CoboCup simulation games [28]; planning schedules for parallel genetic algorithms of which each one solves a sub-problem [14].

Some key issues of future communicative multi-agent systems are the capabilities to co-operate, compete, and co-ordinate their goals, without being handicapped by diverse syntactic and semantic types of heterogeneity. In this work, a planning approach is introduced that considers all these aspects in a single algorithmic methodology. First, the

methodology is discussed in general, thereafter some details of the prototype implementation are given and compared to related work. Finally, the contribution concludes with some remarks.

2. The CEVOP Methodology

The basic idea of the Co-evolutionary Planning (CEVOP) methodology [16] is to provide simple rules for mapping a multi-agent planning problem to a co-evolutionary algorithm, and to get plans for the agents for optimal co-operative and/or competitive working.

Major Concepts of the Multi-agent Planning Domain

The following are some major concepts, which effect multi-agent planning the most (*Table 1*). These concepts have been adopted as major requirements for the CEVOP methodology:

- Agent interaction: Co-operation/competition interrelationships of agents should be modelled.
- Plan variety: For agents with homogeneity/heterogeneity knowledge bases the algorithm should find compatible/contradicting plans.
- Scaleability: All interactions of a multi-agent system should be modelled.
- Focus: Intensity of a specific interaction should be represented in the final plan.
- Plan diversity: The algorithm should be able to generate several alternative plans.

Some co-evolution-related requirements are further needed to improve the search process:

Table 1. Concepts for Co-evolving Plans of Multiple Agents

Concept	Range	Purpose
Interaction	Co-operation	Evolved Plan Sets Should Allow for Resource Sharing
	Competition	Evolved Plan Sets Should Disallow Resource Sharing
Variety	Homogeneous	Plan Sets Evolve According Similar Rules
	Heterogeneous	Plan Sets Evolve According Different Rules
Scalability	2	Two Plan Sets Co-evolve at the Same Time
	n	Multiple Plan Sets Co-evolve at the Same Time
Focus	Great	A Plan set Should be Superior for a Specific Interaction
	Small	A Plan Set Should be Inferior for a Specific Interaction
Diversity	1	Single Final Plan Generated
	n	Alternative Final Plans Generated

- Problem decomposition: The planning problem is decomposed into plan steps and solved step-wise, which is represented by the increased fitness within the algorithm.
- Parallelism: The algorithm can simultaneously evaluate plans within distinct populations.
- Credit assignment: An optimal plan can be generated by evaluating each rule's contribution to a success of the whole plan. This issue should be considered in the design of the chromosome evaluation criteria.

A further discussion of all concepts of the methodology can be found in [18].

Mapping the Concepts Interaction and Variety

In order to consider the above concepts in the co-evolutionary algorithm, a related mapping schema is defined (*Table 2*). The major concepts of the multi-agent planning domain, interaction type and plan diversity, are represented by related combinations of populations within the co-evolutionary algorithm domain.

Table 2. Mapping Interaction and Variety Between the Domains

Multi-agent Planning		Interaction Type	
		Co-operation	Competition
Plan Diversity	Homogeneous	Same Population	Different Populations
	Heterogeneous	Different Populations	Different Populations
		Co-evolutionary Algorithm Domain	

The mapping between the two domains is principally bidirectional. However, in the methodology only one direction is utilised. The generic algorithmic steps of CEVOP are:

```
1)Encoding: The multi-agent planning concepts are mapped
  according the schema.
2)Planning: The planning problem is solved through a co-
  evolutionary algorithm.
3)Decoding: The solutions are interpreted as alternative
  plans for the multi-agent system.
```

Mapping Plan Concepts

Mapping plan concepts onto related constructs of the co-evolutionary computation aims at representing a planning problem of multiple agents as an optimisation problem in the co-evolutionary computation domain (*Table 3*).

Table 3. Mapping Plan Concepts Between the two Problem Domains

Plan	Chromosome
Alternative Plans	Population
Plan Step	Gene
Goal State	Chromosome Evaluation Criterion
Interaction	Chromosome Evaluation Criterion

Any rule of an agent's knowledge base that could potentially be involved in a plan of that agent is interpreted as a plan step. Thus, a chromosome can combine all possible plans of an agent. Alternative plans evolve towards the desired goal state within a population by applying the related chromosome evaluation criteria. Interaction types and strengths between the agents are represented again by chromosome evaluation criteria. For the criteria themselves no further specifications can be made, since they represent the domain-dependent part of a planning problem.

Interrelationships for Co-evolution

The above mappings already determine the interrelationships required for co-evolving the related populations. From the perspective of the co-evolutionary algorithm, a sample schema that includes all reasonable combinations is depicted in (*Figure 1*). It consists of three populations with three interactions.
In this example, each population is assumed to represent a planning problem for several co-operating homogeneous agents. However, each population could also be reduced to represent the plan optimisation process of a single agent that has no homogeneous co-operation interrelationship.

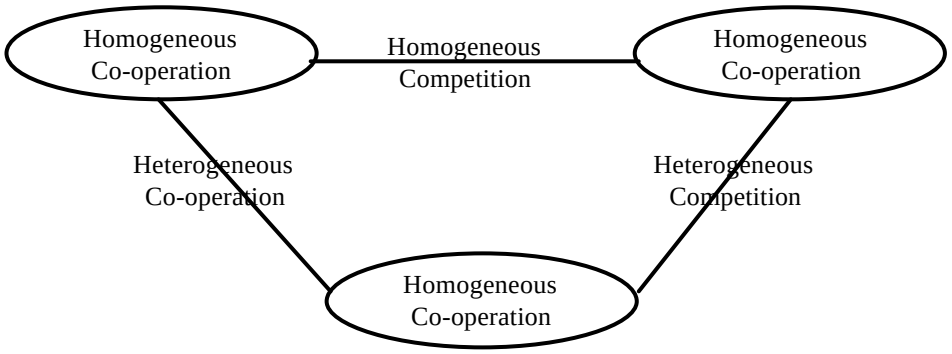


Figure 1: Sample Co-operative and/or Competitive Evolution of three Homogeneous and/or Heterogeneous Plan Populations

In total six different planning problems of multiple agents are solved simultaneously: three homogeneous co-operative, one homogeneous competitive, one heterogeneous competitive, and one heterogeneous co-operative planning problem. However, only three sets of plans are evolved here, where the three interaction types can be considered as further constraints for these plans. As already mentioned in the mapping schema (*Table 2*), homogeneous/heterogeneous refers to the knowledge base of the agents.

The Generic Algorithm for n Plan Populations

The algorithm is introduced here for n agents and the three cases: heterogeneous co-operation, homogeneous competition, and heterogeneous competition. Where, a phenotype represents a single plan consisting of several plan steps. And finally, each gene represents a single plan step.

1. For all n populations
1.1 Randomly pickup plan steps from P_i for population A_i
2. For all n populations
2.1 Cross-over inside A_i
3. For all n populations
3.1 Mutate distance values inside A_i
4. For all n populations
4.1 Calculate Fitness F_i of Population A_i
4.2 For all m_i interaction relationships of population A_i
4.2.1 Calculate fitness F_{ij} for interaction of A_i with A_j ($i \neq j$)
4.2.2 $F_c = F_c + F_{ij}$
4.3 $F_i' = (F_i + F_c / m_i) / 2$ and $F_c = 0$
5. IF fitness F_1 AND ... AND F_m satisfied OR other termination THEN GO TO 7
6. For all n populations
6.1 Fitness selection on A_i
6.2 GO TO 2
7. For all n populations
7.1 PRINT A_i

A_i is the plan population to be optimised. P_i is the set of all possible individual plan steps of agent i. Fitness F_i' of population A_i is determined by accumulating its own fitness F_i and all m fitness values for interrelationships F_{ij}. F_c represents the co-operation and/or competition strengths of agent i with the others.

A fitness function F is a group of domain-specific chromosome evaluation criteria. In each generation it is applied to each chromosome, to calculate the chromosome's fitness value to the desired target state, which represents the final goal of a plan.

The focusing factor that represents interrelation strengths between two agents (*Table 2*) may be encoded in each related fitness function for interrelationship F_{ij}. Which, in turn can be implemented as a chromosome evaluation criterion.

Interpretation of the Algorithm for the Case of Homogeneous Co-operation

In case of k co-operating homogeneous agents, the set of all possible plans P_i is identical for all agents A_{i1}, ..., A_{ik}. Thus, the plans can co-operatively evolve inside a single population. For this purpose the plans of all these agents are represented inside a single chromosome. This case appears in SoccerTeam for the 10 players of a team, except the goal keeper.

3. CEVOP Applied in the SoccerTeam Prototype

In Soccer game, all rules apply equally to both teams. Thus, the teams are homogeneous. The goal of each team is to play together and to shoot the ball into the opponent's goal. Thus, the players of a team co-operate, but the teams compete. Besides the rules that apply to the 10 players of a team, some additional rules apply to the goal keeper. Thus, the 10 players of a team are homogeneous and their goal keeper is heterogeneous.

The Planning Model

SoccerTeam is a prototype [7] with an emphasis on the implementation of the CEVOP methodology. Because of the real-time constraints of the soccer game simulator [4], planning is implemented in two successive steps: first, one time planning through CEVOP, thereafter, possibly several times, executing the plans in a specific match.

Game Strategies

Based on the official soccer game rules, players may apply, individually and co-operatively, various types of strategies successively, in order to shoot a goal. Individual strategies are defined as primitive game strategies and a series of several successive primitive game strategies as a combined game strategy. Further, a set of combined game strategies, with one for each player, as a team plan (*Table 4*) [17]. All combinations of all primitive game strategies opens a search space in which the co-evolutionary algorithm seeks for team plans. This search is driven by the objective function.

Phenotype Representation of a Team Plan

A phenotype represents in total with 11 combined game strategies $S_{p,i}$, one for each player. Where p = 1, ..., 11. Each of them consist of i = 1, 2, 3 primitive game strategies $S_{p,1}$, ..., $S_{p,3}$ with their relative locations $\Delta x_{1,1}$, $\Delta y_{1,1}$, ..., $\Delta x_{11,3}$, $\Delta y_{11,3}$, one for each player (*Table 4*). Each row of the table represents one combined game strategy for one player. Three succeeding columns represent all first primitive game strategies of all 11 players. Thus, one phenotype represents, for a specific game constellation, a team plan solution consisting of 11 plans, one for each player. Where each plan consists further of three plan steps.

Table 4. Phenotype Structure of a *Combined* Game Strategy

$S_{1,1}$	$\Delta x_{1,1}$	$\Delta y_{1,1}$	$S_{1,2}$	$\Delta x_{1,2}$	$\Delta y_{1,2}$	$S_{1,3}$	$\Delta x_{1,3}$	$\Delta y_{1,3}$
	...			...			...	
$S_{11,1}$	$\Delta x_{11,1}$	$\Delta y_{11,1}$	$S_{11,2}$	$\Delta x_{11,2}$	$\Delta y_{11,2}$	$S_{11,3}$	$\Delta x_{11,3}$	$\Delta y_{11,3}$

Since, the goal keeper is heterogeneous to the other 10 team members, the related combined game strategies are transformed to genes of two different chromosome types. Which results in two populations with co-operation interaction.

Objective Function

The definition of the fitness function for the chromosomes is a further critical part of the CEVOP methodology, since they drive the populations towards the desired fitness. Depending on different objectives of the populations, the chromosome evaluation criteria are collected in three major groups: [17]

- General Strategies: e.g. moveInField, physicallyPossible.
- Offensive Strategies: e.g. densityAroundBall, distanceTeamGoalRelative.
- Defensive Strategies: e.g. coverClosest, coverGoal.
- Goalkeeper Strategies: e.g. focusBall.

Generating Plans

The phenotype structure is created two times, one for the offensive team and one for the defensive team. Where, the competition between the teams is represented by four further interactions between the four populations. In the planning process one team always generates offensive plans and the other always defensive plans. After the planning process both, offensive and defensive plans, are stored in the knowledge bases of all 22 players, in order to test the significance of the plans; also against each other. Further details of CEVOP for SoccerTeam can be found in [16].

Simulation Results

In first results following convergence behaviour was observed for most of the simulation runs:

- Offensive Strategies: mostly 100%
- Defensive Strategies: at least 65%
- Goalkeeper Strategies: at least 90%.

The reason for not always converging to 100% is probably due to inappropriate design of the chromosome evaluation criteria. Specifically, for some game constellations some criteria may introduce conflicting objectives, such as coverClosest and coverGoal, in case of the defensive team.

4. Related Work

The following two work on learning Soccer strategies of the RoboCup simulation league are closely related to this approach. The team of Sean Luke [19] learns from sample random competitions by using genetic programming. The team is evolved from genetic programming (Soccer) functions through competitive fitness. It had no success at

RoboCup'97, but won the RoboCup Scientific Challenge Award [13]. Team Darwin United [2] also learns team strategies by genetic programming of agents. It had no success at RoboCup'98. The team is evolved from genetic programming (Soccer) functions through credit assignment.
Both of them have the same objective. The major difference to the CEVOP approach is due to different objectives:

- Since at the team strategy level no domain information is given to the algorithm, an exhaustive search is performed at this level of abstraction; whereas in CEVOP, team strategies are given and the algorithm seeks for strategic game constellations.
- Team strategies are evolved from low-level Soccer strategies and random games; CEVOP uses low-level Soccer strategies, too, but simulates random games through the evolutionary search.
- In general, these approaches showed that evolutionary computation is capable of evolving higher-order concepts through generalisation from primitive strategies of a specific domain. However, the reason why they were not successful in RoboCup competitions is that good Soccer team strategies were already successfully implemented by the teams with analytical approaches.

On the other hand, in a CEVOP run of SoccerTeam several team strategies are already given and the algorithm evolves for sample constellations the best plan solution, which can be interpreted as seeking for optimal applicable team strategies. The objective here is to learn prior to a game, for a given constellation optimal successive team constellations. Which, without CEVOP, had to be sought at run-time in a game.

5. Conclusions

The CEVOP methodology was introduced, which provides an algorithmic solution for co-operative/competitive evolution of homogeneous/heterogeneous plans by mapping a given plan problem to a single search space to be explored by a co-evolutionary algorithm. The algorithm co-evolves multiple plan populations simultaneously and seeks for solutions that satisfy all planning constraints defined by the chromosome evaluation criteria.
This work is still in progress. First simulation results of CEVOP for the Soccer domain show significant results. SoccerTeam is currently implemented in a very simple form that applies just one plan step for each player, in order to inspect the generated plans. An extension of SoccerTeam capable to compete at RoboCup'03 is planned. Source and binaries of the current software are available at the author's Web address.

References

1. Andre, David; 1995; "The automatic programming of agents that learn mental models and create simple plans of action"; IJCAI-95; Morgan Kaufmann
2. Andre, David; Teller, A.; 1999; "Evolving Team Darwin United"; Asada, M.; RoboCup-98; LNCS; Springer; Heidelberg
3. Barone, Luigi; While, Lyndon; 1998; "Evolving Computer Opponents to Play a Game of Simplified Poker"; ICEC'98;

4. Chen, Mao; Foroughi, Ehsan; Heintz, Fredrik; Zahn, Xiangh Huang; Kapetanakis, Spiros; Kostiadis, Kostas; Kummeneje, Johan; Noda, Itsuki; Obst, Oliver; Riley, Pat; Steffens, Timo; Wang, Yi; Yin; Xiang; 2001; "RoboCup Soccer Server"; RoboCup Federation; http:// www.robocup.org/
5. De Jong, Kenneth A.; 1990; "Genetic-algorithm-based learning"; Kodratoff, Y.; Michalski, R. S; Machine Learning; Morgan Kaufmann
6. De Jong, Kenneth A.; Spears, W. M.; 1991; "Learning Concept Classification Rules Using Genetic Algorithms"; 12th International Joint Conference on Artificial Intelligence; Sydney
7. Dix; Jürgen; Muñoz-Avila, Héctor; Nau, Dana S.; 2000; "IMPACTing shop: Planning in a Multi-Agent Environment"; *CL-2000 Workshop on Computational Logic in Multi-Agent Systems*
8. Durfee, E. H.; Lesser, V. R.; 1991; "Partial Global Planning: A Coordination Framework for Distributed Hypothesis Formation"; *IEEE Transactions on Systems, Man, and Cybernetics*
9. Ferber, Jacques; 1999; "Multi-agent Systems - An Introduction to distributed artificial intelligence"; Addison-Wesley; New York
10. Grefenstette, John J.; 1989; "A system for learning control rules with genetic algorithms"; Proc. Third Intl. Conf. Genetic Algorithms; Morgan Kaufmann
11. Grefenstette, John J.; 1991; "Lamarckian learning in multi-agent environments"; Proc. Fourth Intl. Conf. of Genetic Algorithms; Morgan Kaufmann
12. Grefenstette, John J.; 1992; "The evolution of strategies for multi-agent environments"; Adaptive Behavior
13. Hedberg, Sara; 1997; "Robots playing Soccer ? RoboCup poses a new set of AI challenges"; IEEE Expert;
14. Husbands, Philip; Mill, Frank; 1991; "Simulated coevolution as the mechanism for emergent planning and scheduling"; Belew, Richard K.; Booker, Lashon B.; *Proceedings of the Fourth International Conference on Genetic Algorithms*; Morgan Kaufmann
15. Koza, John R.; 1992; "Genetic Programming"; *The MIT Press*; Cambridge
16. Kumova, Bora I.; 2002; "Co-evolutionary Planning"; PhD thesis; Dokuz Eylul University
17. Kumova, Bora I.; 2002; "The Game Staregies of SoccerTeam"; *TAINN*'02; Istanbul University
18. Kumova, Bora I.; Imada, Akira; 2002; "The Concepts of the SoccerTeam Prototype"; submitted to *ICS'02 WS on AI*; National Dong Hwa University; Hualien
19. Luke, Sean; 1998; "Genetic Programming Produced Competitive Soccer Softbot Teams for RoboCup97"; GP98; J. Koza et al.; Morgan Kaufmann; San Fransisco
20. Maturana, F.; Balasubramanian, S.; Norrie, D.H.; 1996; "A Multi-Agent Approach to Integrated Planning and Scheduling for Concurrent Engineering"; Proceedings of the International Conference on Concurrent Engineering: Research and Applications; Toronto
21. Paolucci, M.; Kalp, D.; Pannu, A.; Shehory, O.; and Sycara, K. 2000; "A planning component for retsina agents"; Wooldridge, M.; Lesperance, Y.; Lecture Notes in Artitcial Intelligence; Intelligent Agents VI; Springer
22. Paredis, Jan, Westra, R.; 1997; "Coevolutionary Computation for Path Planning"; Proceedings 5th European Congress on Intelligent Techniques and Soft Computing; H.-J. Zimmermann; Verlag Mainz
23. Pollack, Jordan B.; Blair, Alan D.; 1998; "Co-evolution in the successful learning of Backgammon strategy"; Machine Learning;
24. Reiher, P.; Guy, R.; Yavis, M.; Rudenko, A.; 2000; "Automated Planning for Open Architectures"; OpenArch '2000
25. Reynolds, Craig W.; 1994; "Competition, coevolution and the game of tag"; Brooks, Rodney A.; Maes, Pattie; Fourth International Workshop on the Synthesis and Simulation of Living Systems; MIT Press; Cambridge
26. Rosin, Christopher Darrell; Belew, Richard K.; 1997; "New methods for competitive coevolution"; Evolutionary Computation

27. Wooldridge, Michael; Jennings, Nicholas R.; 1999; "The Co-operative Problem Solving Process"; Journal of Logic & Computation
28. Yunpeng, Cai; Jiang, Chen; Jinyi, Yao; Shi, Li; 2001; "Global Planning from Local Eyeshot: An Implementation of Observation-base Plan Coordination in CoboCup Simulation Games";