Community Swarm Optimization

Rawan Ghnemat[1], Cyrille Bertelle[1], and Gérard H.E. Duchamp[2]

[1] LITIS - University of Le Havre,
25 rue Philippe Lebon - BP 540
76058 Le Havre cedex, France
rawan.ghnemat@litislab.eu
cyrille.bertelle@litislab.eu

[2] LIPN - University of Paris XIII,
99 avenue Jean-Baptiste Clément
93430 Villetaneuse, France
ghed@lipn.univ-paris13.fr

Summary. The development of distributed computations and complex systems modelling lead to the creation of innovative algorithms based on interacting virtual entities, specifically for optimisation purpose within complex phenomena. Particule Swarm Optimisation (PSO) and Ant Colony Optimisation (ACO) are two of these algorithms. We propose in this paper a method called Community Swarm Optimisation (CSO). This method is based on more sophisticated entities which are defined by behavioral automata. This algorithm leads to the emergence of the solution by the co-evolution of their behavioral and spatial characteristics. This method is suitable for urban management, according to improve the understanding of the individual behaviors over the emergent urban organizations.

Key words: swarm optimisation, community dectection, self-organization, automata, evolutive methods, geographic systems, ressource management.

1 Introduction

Artificial complex systems allow to implement distributed problems solving. Such systems are composed of interacting entities from where emergent properties appear. We focus, in this paper, on the artificial swarms or populations methods allowing these emergent property computations. The artificial swarms or populations move on a representation of the environment or on a representation of the space of the solutions. The emergent properties of these artificial systems are the collective meaning of the system itself, according to some objective functions. In many cases, we can express the control of the system by these objective functions as an optimization problem; the optimal configuration could be expected in advance or could be computed during the evolution of the system to reach some adaptive properties.

In the following, we will present some of these modern methods concerning artificial swarm intelligence and we will propose a new one, called Community Swarm Optimization. This method is mainly based on the concept of spatial evolutive populations of behavioral entities. The concept of community is the basic property of this method.

Definition 1. (Community operational definition)
A community is a system or an organization which is characterized by a spatial property, a behavior property and the interaction between the both.

Example 1. In ecology, a community is a group of plants or animals living in a specific region and interacting with one another.

Example 2. The spatial patterns generated by Schelling's segregation models [15] are some examples of communities and these spatial patterns are linked with some elementary behavioral rules implemented for each grid case. These rules describe, for each step, the movement of each individual according to its neighborhood.

In SCO method, we need to represent an efficient way to describe the behavior of each entity and we use the algebraic structures called automata with multiplicities [16]. The main advantage of these automata is to be associated to algebraic operators leading to automatic computation. With these operators, we can define behavioral distances for the entities modelled with these automata. The behavioral distance is one of the major keys of this new method. The section 3 describes the algebraic basis for the automata management used in this method. In the section 4, we describe the proposed method and in the section 5, we discuss about some applications which can be efficiently modelled by this method, according to their own complexity.

2 Swarm Optimisation Methods

Decentralized algorithms have been implemented since many years for various purposes. In this algorithm category, multi-agent systems can be considered as generic methods [19]. Agent-based programming deals with two main categories of agent concepts: cognitive agents and reactive agents. The first category concerns sophisticated entities able to integrate, for example, knowledge basis or communications systems. Generally, efficient computations, based on these cognitive architectures, implement few agents. The second category of agents, based on reactive architecture, is expected to be used inside numerous entity-based systems. The goals of programs using such architectures, is to deal with emergent organizations using specific algorithms called emergent computing algorithms. Swarm Intelligence is the terminology used to point out such reactive agent-based methods where each entity is built with the same basis of behavior, but react in autonoumous way. Swarm Optimization methods concern the problems of optimization where the computation of a function extramum is based on the concept of the swarm intelligence.

Ant Colony Optimization (ACO) methods [4] is a bio-inpirated method family where the basic entities are virtual ants which cooperate to find the solution of graph-based problems, like network routing problems, for example. Using indirect communications, based on pheromons deposite over the environment (here a graph), the virtual ants react in elementary way by a probabilistic choice of path weighted with two coefficients, one comes from the problem heuristic and the other represent the pheromon rate deposit by all the ants until now. The feed-back process of the whole system over the entities is modelled by the pheromon action on the ants themselves.

Particule Swarm Optimization (PSO) is a metaheuristic method initially proposed by J. Kennedy and R. Ebenhart [12].This method is initialized with a virtual particles set which can move over the space of the solutions corresponding to a specific optimization problem. The method can be considered like an extension of a bird flocking model, like the BOIDS simulation from C.W. Reynolds [14]. In PSO algorithm, each virtual particles moves according to its current velocity, its best previous position and the best position obtained from the particles of its neighborhood. The feed-back process of the whole system over the entities is modelled by the storage of this two best positions as the result of communications between the system entities.

Other swarm optimization methods have been developped like Artificial Immune Systems [6] which is based on the metaphor of immune system as a collective intelligence process. F. Schweitzer proposes also a generic method based on distributed agents, using approaches of statistical many-particle physics [17].

The method proposed in this paper, is called Communities Swarm Optimization (CSO) and it consists in the co-evolving of both the spatial coordinates and the behavior of each individual of a virtual population of automata. The feed-back process of the whole system over the entities is modelled by a genetic algorithm based on this co-evolving. The automata behaviors allow to define for each individual, a set of some arbitrary complex transition rules. We develop the formalism needed to describe this method and the associated algorithm in the two next sections.

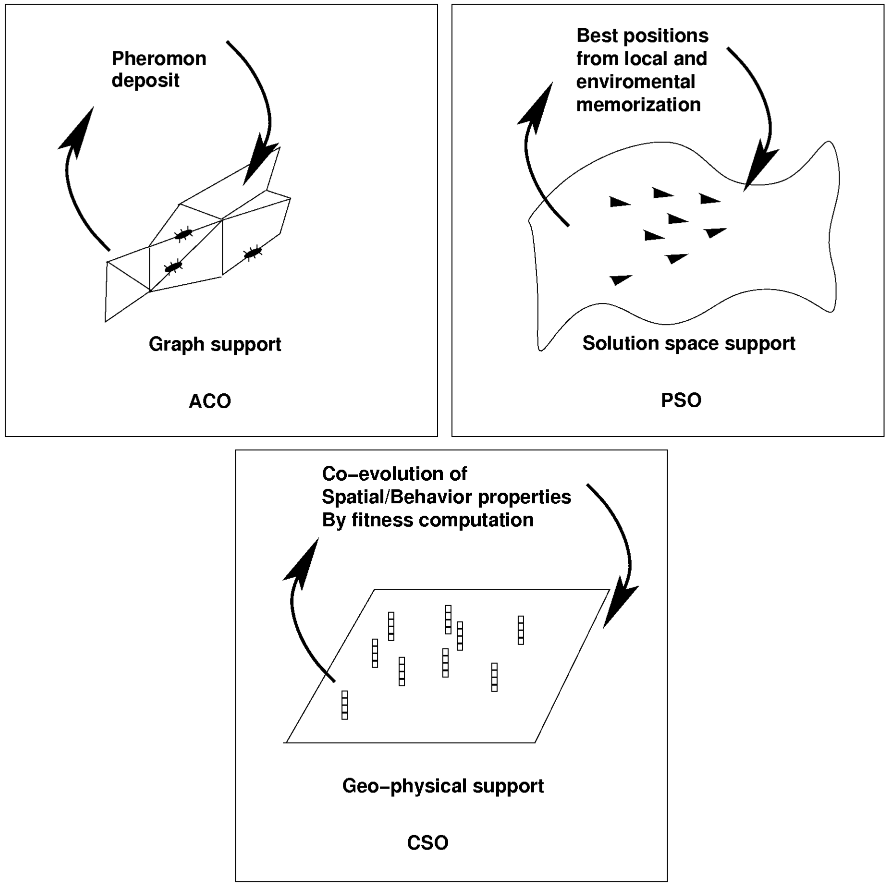

Fig. 1. Support and feed-back comparison from Ant Colony Optilization (ACO), Particule Swarm Optimization (PSO) and Community Swarm Optimization (CSO)

3 Spatial Behavioral Automata

3.1 Behavior modelling using automata

An automaton with multiplicities is an automaton with output values belonging to a specific algebraic structure, a semiring, including real, complex, probabilistic, non commutative semantic outputs (transducers) [10, 18]. In that way, we are able to build effective operations on such automata, using the properties of the algebraic structures where belong the output data. We are specifically able to describe automata by means of a matrix representation with all the power of the new (i.e. with semirings) linear algebra.

Definition 2. (Automaton with multiplicities)
An automaton with multiplicities over an alphabet A *and a semiring* K *is the 5-uple* (A, Q, I, T, F) *where*

- $Q = \{S_1, S_2 \cdots S_n\}$ *is the finite set of state;*
- $I : Q \mapsto K$ *is a function over the set of states, which associates to each initial state a value of* K*, called entry cost, and to non- initial state a zero value ;*
- $F : Q \mapsto K$ *is a function over the set states, which associates to each final state a value of* K*, called final cost, and to non-final state a zero value;*
- T *is the transition function, that is* $T : Q \times A \times Q \mapsto K$ *which to a state* S_i*, a letter* a *and a state* S_j *associates a value* z *of* K *(the cost of the transition) if it exist a transition labelled with* a *from the state* S_i *to the state* S_j *and and zero otherwise.*

Remark 1. We have not yet, on purpose, defined what a semiring is. Roughly it is the least structure which allows the matrix "calculus" with unit (one can think of a ring without the "minus" operation). The previous automata with multiplicities can be, equivalently, expressed by a matrix representation which is a triplet

- $\lambda \in K^{1 \times Q}$ which is a row-vector which coefficients are $\lambda_i = I(S_i)$,
- $\gamma \in K^{Q \times 1}$ is a column-vector which coefficients are $\gamma_i = F(S_i)$,
- $\mu : A^* \mapsto K^{Q \times Q}$ is a morphism of monoids (indeed $K^{Q \times Q}$ is endowed with the product of matrices) such that the coefficient on the q_ith row and q_jth column of $\mu(a)$ is $T(q_i, a, q_j)$

The figure 2 describes the linear representation of a probabilistic automaton which is a specific automaton where output values are probabilistic values. For these probabilistic automata, the sum of the coefficients of each matrix lines is equal to 1.

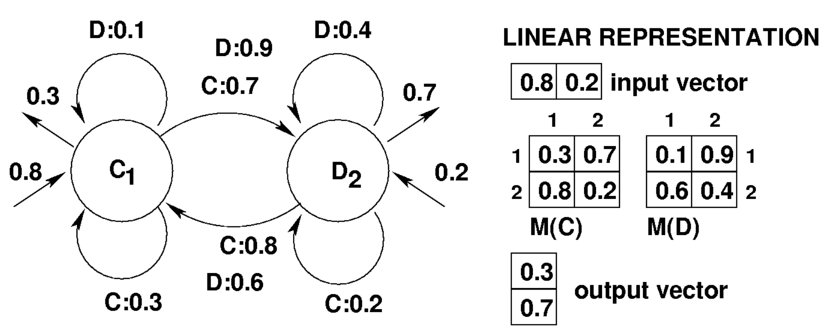

Fig. 2. Probabilistic automata over the alphabet {C,D} and its linear representation

Definition 3. (Automata-Based Agent Behavior)
We represent the agent behavior by automata with multiplicities (A, Q, I, T, F) *over a semiring* K*:*

- *The agent behavior is composed of a states set* Q *and of rule-based transitions between them. These transitions are represented by* T*;* I *and* F *represent the initial and final transitions;*
- *Alphabet* A *corresponds to the agent perceptions set;*
- *The semiring* K *is the set of agent actions, eventually associated to a probabilistic value which is the action realization probability (as defined in [8]).*

3.2 Spatial Automata and associated spatial distance

Definition 4. (Spatial Automata-Based Agent)
A spatial automata-based agent is defined by its structural representation:

- *An automaton with multiplicities corresponding to its behavior as a whole processus managing its perceptions and its actions over its environment. They include its communication capabilities and so its social behavior;*
- *A spatial location defined on some specific metric space.*

Remark 2. According with this previous definition, we define two metrics on the spatial automata-based agent. The first one concerns a spatial distance which is directly induced by the metric of the spatial location (from any standard Hölder norm). The second one is more innovative and concerns a behavioral distance or semi-distance. One of the major interest of the previous automata-based modelling is to be able to define such behavioral distance which lead to powerful automatic processes dealing with self-organization. We detail this definition in the next section.

3.3 Metric spaces for behavioral distances

The main advantage of automata-based agent modelling is their efficient operators. We deal is this paragraph with an innovative way to define behavioral semi-distance as the essential key of the swarm algorithm proposed later.

Definition 5. (Evaluation function for automata-based behavior)
Let x *an agent whom behavior is defined by* $\mathcal{A}$*, an automaton with multiplicities over the semiring* K*, we define the evaluation function* $e(x)$ *by:*

$$e(x) = V(\mathcal{A})$$

where $V(\mathcal{A})$ *stands for the vector of all coefficients of* (λ, μ, γ)*, the linear representation of* $\mathcal{A}$*, defined in remark 1.*

Definition 6. (Behavioral semi-distance)
Let x and y two agents and $e(x)$ and $e(y)$ their respective evaluations as described in the previous definition 5. We define $d(x, y)$ a semi-distance or pseudometrics [3] *between the two agents x and y as*

$$d(x, y) = ||e(x) - e(y)||$$

a vector norm of the difference of their evaluations.

3.4 Genetic operators on spatial automata-based agent

We consider in the following, a population of spatial automata-based agents, each of them is represented by a chromosome, following the genetic algorithm basis. We define the chromosome for each spatial automata-based agent as a couple of two sequences:

- the sequence of all the lines of the matrices from the linear representation of the automata. The matrices, associated to each letter from the alphabet of the agent perceptions, are linearly ordered by this alphabet and we order all the lines following these matrices order [3]. The figure 3 describes how this sequence is created from a two matrices linear representation;
- the sequence of all its spatial coordinates.

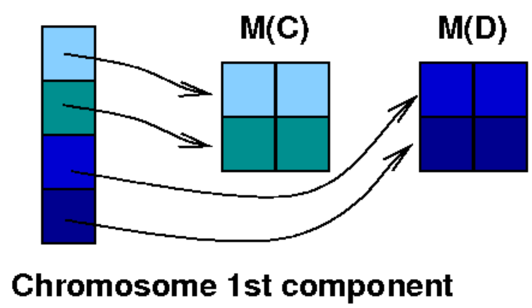

Fig. 3. Chromosome first component building from the matrix lines of the linear representation of an automaton over the alphabet {C,D}

In the following, genetic algorithms are going to generate new automata containing possibly new transitions from the ones included in the initial automata.

The genetic algorithm over the population of spatial automata-based agent follows an evolutive iteration composed of two main operators, as on adaptation of the classical genetic operators [11]:

[3] see [5] ch IX

- *Reproduction (Duplication and Crossing-over)*: This operator is a combination of the standard duplication and crossing-over genetic operators. For each couple of spatial automata (called the parents), we generate two new spatial automata (called the children) as the result of the chromosomes crossing and we keep, without change, the parent spatial automata. To operate for the crossing-over operation, we have to compute
  - the automata of the behaviors of the two children. For this purpose, we consider a sequence of lines for each matrix of the linear representation of one of the two parents and we make a permutation on these chosen sequences of lines between the analogue matrix lines of the other parent;
  - the spatial locations of the two children. These children locations can be choosen from many ways: on the linear segment defined by the parent locations or as the nodes of the square obtained with the parent and the children as describe in the figure 4.
- *Mutation*: This operator deals only with the linear representation of the spatial automata-based agent. With a low probability, each matrix lines from this linear representation is randomly chosen and a sequence of new values is given for this line (respecting some constraints if exist, like probabilistic values [3]).

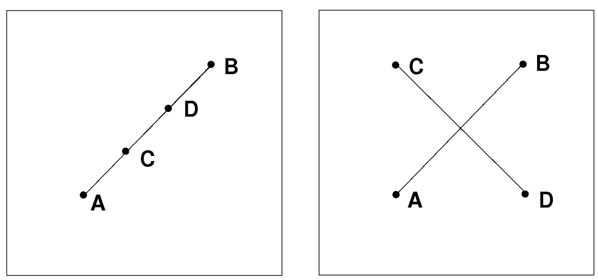

Fig. 4. Spatial locations for the children C and D from the parents A and B, after a reproduction step. Two possible locations computation are presented in the two sub-figures.

4 Community Swarm Optimization Algorithm

4.1 Adaptive objective function for community-based swarm optimization

The community swarm optimization method is based on a genetic algorithm over a population of spatial automata-based agents. The formation of the community is the result of the population evolution, crossing by a selection process computed with the fitness function defined in the following.

For this computation, we deal with two distances defined on agent set. The first is the spatial distance associated to the agent spatial location and the second is the behavioral semi-distance defined in the definition 6.

Definition 7. Community clustering and detection fitness
Let $\mathcal{V}_x$ *a neighbourhood of the agent* x*, relatively to its spatial location. We define* $f(x)$ *the agent fitness of the agent* x *as :*

$$f(x) = \begin{cases} \dfrac{card(\mathcal{V}_x)}{\displaystyle\sum_{y_i \in \mathcal{V}_x} d(x, y_i)^2} & \text{if } \sum_{y_i \in \mathcal{V}_x} d(x, y_i)^2 \neq 0 \\ \infty & \text{otherwise} \end{cases}$$

where $d(x, y)$ *is the behavioral semi-distance between the two agents* x *and* y*.*

This fitness allows to implement a co-evolution process which generates an emergent set of community swarms. These community swarms formation is the result of both the emergence of the spatial location of the generated communities and the adaptive behavior of the communities as the result of the homogeneisation of the behavioral automata of all the agent which composed these emergent communities.

4.2 General CSO algorithm

CSO algorithm needs a initial step description which is the major step of the modelling process. The way of going from the problem formulation to the initial spatial automata-based agents must be realized in accuracity. The formal description of the methodology to use, for this initial step, is described in Algorithm 1.

Algorithm 1: Methodology to model the initial step of CSO

1. Problem formulation by the definition of a set of transition rules ;
2. Building of the behavioral automata based on the previous set of transition rules, describing the sequences and the context of their applications ;
3. Discretization of the spatial domain, according to its topological properties (Cellular automaton, network or graph, Geographical Information System) with the spatial location of the initial virtual population of spatial automata-based agents;

The core of the CSO algorithm is described by the iterative scheme defined in the Algorithm 2.

Algorithm 2: Iteratice scheme of CSO

Building the initial virtual population of the spatial automata-based agents (following the methodology of Algorithm (1)) ;
repeat
  for *Each couple of individuals in the population* **do**
    Reproduction step generating 2 new children as described in the section (3.4) ;
    Mutation step as described in the section (3.4) ;
  Selection of the half population of the individuals corresponding to the highest values of the agent fitness described in section (4.1) ;
until *(the sum of the fitness values of the whole population reachs a threshold) or (the maximum iteration number is reached)* ;

An example of fitness function computation output is illustrated with the figure 5 where we show, on the same population, an high level fitness individual which will be probably kept inside the population at the next iteration and a low level fitness individual which be probably removed from the population at the next iteration. The color used to graphically described the chromosome composition, allows to appreciate the similarity of the individuals.

5 Conclusion and Perspectives

In this paper, we describe Community Swarm Optimization (CSO) method which can be described as a swarm optimization process. With the comparison of other methods from this category (ACO and PSO), CSO differs mainly on the modelling purpose. CSO deals with transition rules included in data structures (automata with multiplicities) for which algebraic operators allow to implement automatic computation for self-organizational phenomena.

Presented as a very generic method in this paper, CSO can be efficiently applied in engineering problems where the spatial characteristics are not only additional coordinates for the data but the spatial organization is the result of the self-organization process as the output of the method. Typically, urban and territorial management need this kind of modelling to improve the decision making within a sustainable development (including environmental, economic and social apects).

Schelling's Segregation model [15] is one of the basic problem dealing with spatial self-organizations within urban or territorial management. In this model, the spatial domain, based on cellular automata, is constrained and the rules are elementary. Another example concerns partitionning problem for territorial management which can be computed by genetic algorithms [20], using objective functions with various parameters to adjust. Following

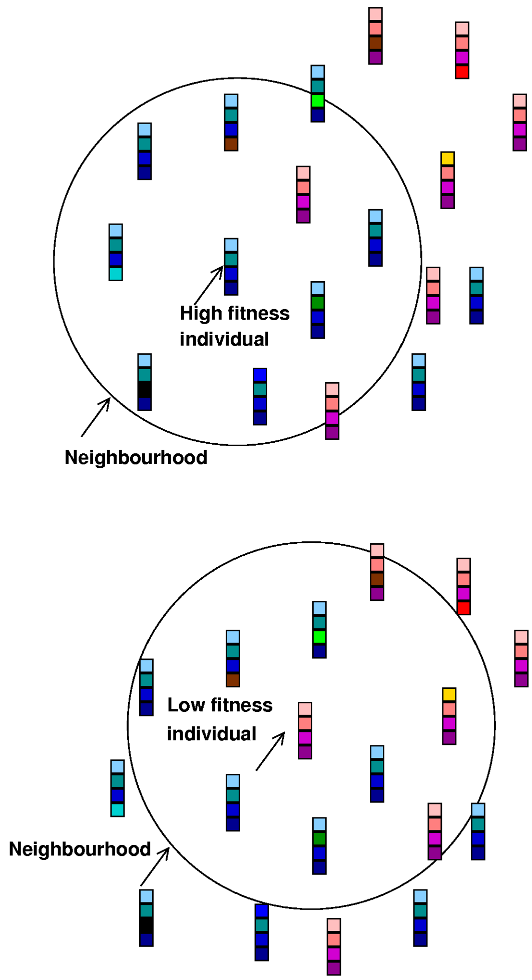

Fig. 5. Two examples of the resultat of the selection operator within the CSO algorithm

Benenson and Torrens [1], we have to deal nowadays with the understanding of the development of urban dynamics as collective phenomena. Models of election/voting (studying the political affiliation of citizen under the influence of their environment and neighbors), diffusion of innovation in the city (studying the acceptation or rejection of innovation from citizen depending on their environment and neighbors) are typically spatial complex systems with multi-dimensional (social, economic, ...) rule-based aspects which need accurate transition machines and efficient self-organization processes that CSO is expected to solve.

References

1. Benenson, I., Torrens, P.M. (2004) *Geosimulation - Automata-based modeling of urban phenomena*, Wiley.
2. Berstel, J., Reutenauer, G. (1988) *Rational series and their languages*, EATCS.
3. Bertelle, C., Flouret, M., Jay, V., Olivier, D., Ponty, J.-L. (2001) *Genetic algorithms on automata with multiplicities for adaptive agent behaviour in emergent organizations* In *SCI'2001*, Orlando, Florida, USA, 22-25th July 2001.
4. Bonabeau, E., Dorigo, M., Theraulaz, G. (1999) *Swam Intelligence, from natural to artificial systems*, a volume in the Santa Fe Institute Studies in the Sciences of Complexity, Oxford University Press.
5. Bourbaki, N. (1998) *Elements of Mathematics: General Topology*, Chapters 5-10, Springer-Verlag Telos.
6. De Castro, L.N., Timmis, J. (2002) *Artificial immune system: a new computational approach*, Springer-Verlag, London, U.K.
7. Edmonds, B., Hales, D. (2005) *Computational Simulation as Theoretical Experiments*, Journal of Mathematical Sociology, Vol. 29(3):209-232.
8. Ghnemat, R., Oqeili, S., Bertelle, C., Duchamp, G.H.E. (2006) *Automata-Based Adaptive Behavior for Economic Modelling Using Game Theory*, in Aziz-Alaoui, M.A., Bertelle, C., *Emergent Properties in Natural and Artificial Dynamical Systems*, Springer.
9. Ghnemat, R., Bertelle, C., Duchamp, G.H.E. (2007) *Adaptive Automata Community Detection and Clustering, a generic methodology*, in *Proceedings of World Congress on Engineering 2007, International Conference of Computational Intelligence and Intelligent Systems*, pp 25-30, London, U.K.
10. Golan, J.S. (1999) *Power algebras over semirings*, Kluwer Academic Publishers.
11. Goldberg, D.E. (1989) *Genetic Algorithms*, Addison-Wesley.
12. Kennedy, J., Eberhart, R.C. (1995) *Particle Swarm Optimization* In *Proceedings of IEEE International Conference on Neural Networks (Perth, Australia)*, IEEE Service Center, Piscataway, NJ, 5(3), pp 1942-1948.
13. Le Moigne, J.-L. (1999) *La modélisation des systèmes complexes*, Dunod.
14. Reynolds, C.W. (1987) *Flocks, Herds and Schools: a distributed behavioral model* In *Computer Graphics*, 21(4) (SIGGRAPH'87 Conference Proceedings), pp 25-34.
15. Schelling, T.C. (1971) *Dynamic Models of Segregation*, Journal of Mathematical Sociology, Vol. 1:143-186.
16. Schützenberger, M.P. (1961) *On the definition of a family of automata*, Information and Control, Vol. 4:245-270.
17. Schweitzer, F. (2003) *Brownian Agents and Active Particles*, Springer.
18. Stanley, R.P. (1999) *Enumerative combinatorics*, Cambridge University Press.
19. Weiss, G. (ed.) (1999) *Multiagent Systems*, MIT Press.
20. Xiao, N. (2005) *Geographic optimization using evolutionary algorithms* In *8th International Conference on GeoComputation*, University of Michigan, USA.