

# A classification scheme for agent based approaches to dynamic optimization

Adil Baykasoglu · Zeynep D. U. Durmusoglu

Published online: 3 January 2012  
© Springer Science+Business Media B.V. 2011

**Abstract** Several papers in the literature employ agent-based modeling approach for providing reasonable solutions to dynamic optimization problems (DOPs). However, these studies employ a variety of agent-based modeling approaches with different strategies and features for different DOPs. On the other hand, there is an absence in the literature of a formal representation of the existing agent-based solution strategies. This paper proposes a representation scheme indicating how the solution strategies with agent-based approach can be summarized in a concise manner. We present these in a tabular form called “Agent Based Dynamic Optimization Problem Solution Strategy” (ABDOPSS). ABDOPSS distinguishes different classes of agent based algorithms (via communication type, cooperation type, dynamism domain and etc.) by specifying the fundamental ingredients of each of these approaches with respect to problem domain (problems with dynamic objective functions, constraints and etc.). This paper also analyzes 18 generic studies in the literature employing agent-based modeling based on ABDOPSS.

**Keywords** Dynamic optimization problems (DOPs) · Agent-based modeling · Agent based systems · Agent based dynamic optimization problem solution strategy (ABDOPSS)

## 1 Introduction

Optimization efforts through system modeling have been one of the most convenient approaches employed by many researchers to obtain promising solutions for several problems. Nonetheless, optimizing the dynamic systems enclosing continuous change including the change in objective function(s), constraint/restriction(s) and parameter(s) has been a difficult task for the researchers.

---

A. Baykasoglu (✉)  
Department of Industrial Engineering, Dokuz Eylul University, Izmir, Turkey  
e-mail: baykasoglu@gmail.com

Z. D. U. Durmusoglu  
Department of Industrial Engineering, University of Gaziantep, Gaziantep, Turkey

On the other hand, some of the researchers studying operations research (OR) have attempted to solve these problems by using the well-known traditional optimization techniques like linear programming and integer programming (Baykasoglu and Unutmaz Durmusoglu 2011). However, solutions obtained for a certain time that is desirable or optimal for that time slot, often have not be preferable for another time slot. Furthermore, obtaining exact solutions using mathematical methods for each time slot has been unaffordable or even sometimes has not ended-up with feasible solutions. Therefore, notwithstanding its widespread use, the static optimization approach has increasingly approached its limits (Borst et al. 2005). This fact certainly inspired many researchers to employ agent-based modeling for the optimization of dynamic systems. As indicated by Jung et al. (2011), the agent paradigm has been shown to be a promising approach to develop intelligent, heterogeneous, and open systems, because of its features, such as *autonomy*, *flexibility*, and *cooperative problem solving behavior* (Jung et al. 2011). In this regard, reusable agents have provided many opportunities and they saved researchers from moving beyond reinventing, representing, and re-implementing the problem and thereby the cost of providing solutions has been decreased (Neches et al. 1991).

Agent-based models addressing dynamic optimization problems (DOPs) have also allowed several optimization mechanisms to track the moving optima in the search space. This fact certainly inspired many researchers to employ agent-based modeling for the optimization of dynamic systems.

Even though there have been several papers in the literature employing agent-based modeling approach to provide significant solutions for DOPs, each of these existing studies employs different agent-based modeling approaches having different strategies and agent features. Therefore, currently, we don't have a common successful methodology for the design of agents (Corchado et al. 2008). With dependability and safety in mind, it is vital that the mechanisms for representing and implementing agents are clear and consistent (Fisher et al. 2007). In this regard, it is certain that there is a vacancy in the standard representation and classification of agent-based solution approaches employed for DOPs.

A previously developed scheme (Calégari et al. 1999) for evolutionary algorithms for combinatorial optimization has also inspired us to prepare a standard representation format. In this perspective, one of the objectives of this paper is to present the fundamental features/ingredients of DOPs solved by agent-based modeling. The features positioned in a standard scheme provide an opportunity to present DOPs in a standard way. It can succeed further and in future studies these representation forms can be used to study the role of these features and their importance for solution quality. Thereby, this study provides a basis to analyze the best ways of implementing agent-based approaches to the DOPs. In this perspective, this paper presents an attempt to develop a notion for DOPs solved by agent-based approach. The paper shows how solution strategies with agent-based approach for DOPs can be summarized in a concise manner by informing about the agents employed and the key elements of DOP's. A classification scheme is introduced and presented in a tabular form called agent based dynamic optimization problem solution strategy (ABDOPSS). ABDOPSS distinguishes between different classes of agent based algorithms (via communication type, cooperation type, dynamism domain and etc.) by enumerating the fundamental ingredients of each of these agent-based approaches. At the end, possible uses of the ABDOPSS are illustrated and exemplified for some agent based approaches.

The paper is organized as follows. Sections 2 and 3 introduces the features listed as the ingredients of ABDOPSS. Section 2 specifically focuses on the features of typical DOPs. Subsequently, Sect. 3 introduces the generic features of agent-based approaches. In Sect. 4, some typical agent-based solution approaches that are implemented for DOPs are described

by using the ABDOPSS. Section 5 illustrates the application of ABDOPSS for 18 relevant articles collected from the literature. Finally, Sect. 6 presents concluding remarks with the possible benefits of the proposed classification scheme.

## 2 Problem specific features: a typical dynamic optimization problem

A dynamic problem is considered to be a problem that changes some or all of its characteristics in time (Lung and Dumitrescu 2009). Thereby, optimization problems that are subject to changes, belongs to the class of dynamic optimization problems (DOPs) (Allmendinger and Knowles 2010). In other words; DOPs are “*optimization problems whose optimal solution changes over time during the optimization, which could result from change of environmental parameters, change of constraints, change of objectives and change of problem settings (representations)*” (Jin 2004).

All of the changes that are mentioned above take place in the dynamic environment and these changes create unpredictability even chaos in some situations. These changes can be opportunities improving the solutions or sometimes they can be threats decreasing level of desirability of a solution.

There are different approaches in the literature that is classifying these changes in the dynamic environments. Yang (2007) defines environmental change as: cyclic and random. In cyclic change of environments, with the time going, an old environment reappears exactly (Yang 2007). A cyclic change covers the same fixed length sequence of contraction and/or growth is repeated over time (Abbas et al. 2004). The pattern repeats itself indefinitely (Abbas et al. 2004). However, in random change, there is no information about the structure of change. Random change is also called as acyclic and in random change there are no repeated patterns over the time frame; that is, there is an indefinite random sequence of contractions, growths, and random changes (Abbas et al. 2004).

All of these changes in dynamic environment cause significant changes in model’s structure, objective function, restrictions/constraints, parameters, and in the decision variables. Bui et al. (2011) describe three of these changes through the examples as presented below.

Time-varying objective functions: For example enemy units arrive at a location, making some parts of the objective more difficult. The objective function is not constant over time. Therefore, the objective value of a solution can be different at different times. This usually causes the occurrence of new optima (Bui et al. 2011).

Time-varying variables: An example problem of this category is dynamic machine scheduling where unexpected new jobs arrive. A time-varying variable can be used for determination of the objective value, but can be a late addition or change (Bui et al. 2011).

Time-varying constraints: For example the precedence relationship of tasks. The objective function and variables do not change for this category; however, the constraints may change over time. This category of change does not change the fitness landscape driven by the objective function, but it will affect the areas of feasibility. It is interesting to note that this category of change has not been analyzed in detail in comparison to the other two categories (Bui et al. 2011).

Cruz et al. (2010) provides a valuable survey of research done on optimization in dynamic environments over the past decade. In addition to the content of their paper, they provide a web site (Cruz et al. 2011) and present characteristics of the DOPs with respect to objective function or the restrictions change with time.

We have considered the study of Abbas et al. (2004) to illustrate correspondence between the change types and the changes in the modeling components. Abbas et al. (2004) defines

**Table 1** Change types for a dynamical environment and their corresponding affects in the mathematical model

Change state	Changes defined by (Abbas et al. 2004)	Corresponding change in the model	An example from dynamic production facility
Change 1	Change in the model	Objective function change, Constraint change, Number of decision variables, Type of parameters	Objective of the system can be altered from minimizing the cycle time to minimizing number of tardy jobs
Change 2	Change in the number of concepts	Objective function change, Constraint change, Number of decision variables	New machines can arrive to the system or some machines can be removed from the system
Change 3	Change in the number of features	Constraint change, Objective function change, Number of decision variables	New product orders can arrive to the system requiring different operations
Change 4	Change in the level of noise	Objective function change, Constraint change	Some customers can broke the deal and cancel their orders
Change 5	Change in the class distribution	Parameter change	In case of economic boom, arrival of jobs to the system can be more frequent
Change 6	Change in the sample bias	Parameter change	Since processing times for machines are estimated using sampling, an unexpected sampling error can end up with the change in actual processing times

six common kinds of changes for a typical system having certain data flow structure. These changes are: change in the model, change in the number of concepts, change in the number of features, change in the level of noise, change in the class distribution, and change in the sample bias.

Table 1 both summarizes the change states defined by Abbas et al. (2004) and their corresponding change in a typical mathematical model which is similar to the one defined by Cruz et al. (2010). Examples of each change categories have also been provided through the possible changes in a dynamic production facility. According to the example case, a typical production facility that is operating in a dynamic environment faces with several expected and unexpected changes with the above given change states.

*Change 1:* Change in the model: Change in objectives of a system may lead a significant “change in the model”. Think of the production facility trying to minimize the cycle time and adjusting all of its production strategy with respect to cycle time minimization. If the cost of tardy job increases significantly and the managers will be obligated to change their objective as the minimization of the number of tardy jobs. In this case, the objective function, constraints and the number of decision variables along with the type of parameters may change.

*Change 2:* Change in the number of concepts: Sometimes, concept classes may not be as clear-cut as initially thought so that additional classes may arise or old classes may vanish over time (Abbas et al. 2004). In case of change in the number of concepts, resources used by the system can increase or decrease. New machines can arrive to the system or some machines

can be removed from the system because of the oldness or in functionality. This change can affect the model by changing the variables used in objective function and constraints. This type of change is not expected to create structural change. It is a kind of reconsideration of the model with added/removed variables that is significant for decisions.

*Change 3:* Change in the number of features: In some problems, the number of available features characterizing a problem instance may vary over time. Additional information may become available or new tools may be developed that allow more accurate classification (Abbas et al. 2004). Orders received for new product types are a typical example of this change. In the face of such a change, a production facility may face with new operations required and this can lead changes in objective function, constraints and the number of decision variables.

*Change 4:* Change in the level of noise: The noisiness of the data may change as well (Abbas et al. 2004). This type of change is particularly common when dealing with data coming from sensors. For example, acoustic data collected in an open area can have different noise levels based on the state of the environment (Abbas et al. 2004). In this respect, in the production facility example, if some customers broke the deal and cancel their orders for a production facility, the plans unfortunately changes and it may cause change in objective function and constraints.

*Change 5:* Change in the class distribution: Abbas et al. (2004) exemplifies the change in the class distribution with SARS virus. When the SARS outbreak took place, the class distribution changed almost every day with more healthy people becoming infected effectively increasing the proportion of positive cases. Similar to SARS example, in case of an economic boom, arrival of some certain jobs to the system can be more frequent and this leads change in parameters.

*Change 6:* Change in the sample bias: Since system parameters like processing times for machines are estimated by the sampling of several actual processes, an unexpected sampling error can end up with the change in actual processing times and/or system arrival times.

All those changes defined above also directly effects the unpredictability of the problem domain. Level of unpredictability is vital to determine a proper agent-based solution strategy. Although there are several efforts to match the corresponding changes with the unpredictability, they are still in their infancy. Therefore, there is a serious need for measuring the level of unpredictability of the DOPs.

The relation between the corresponding changes in constraints, objective function and the both with uncertainty and dynamism has been previously presented by (Cruz et al. 2010) in the web site (Cruz et al. 2011) as shown in Fig. 1. While their framework as shown in Fig. 1 does not cover the variability in parameters and variables; it can still be a good reference for future studies. Considering parameter variability and variability in the number of decision variables will certainly increase the complexity of such an analysis.

It is also remarkable that dynamism is one of the components of environmental uncertainty. Indeed, as presented in Fig. 2, environmental uncertainty has three dimensions: dynamism, heterogeneity, and hostility (Newkirk and Lederer 2006). Dynamism, as being the one dimension of the environmental change, (Teo and King 1997) found to have two dimensions. The dimensions are referred to here as changeability (i.e., concerned with the rate of obsolescence and of technology change) and unpredictability (i.e., concerned with competitors' moves and demand changes). The rigor of that study and relative recency of the findings motivated the use of those two dimensions in the current research (Newkirk and Lederer 2006). In this regard the above defined six different changes cover different level of changeability and unpredictability. Categorizing changes into these two dimensions is a topic of future research.

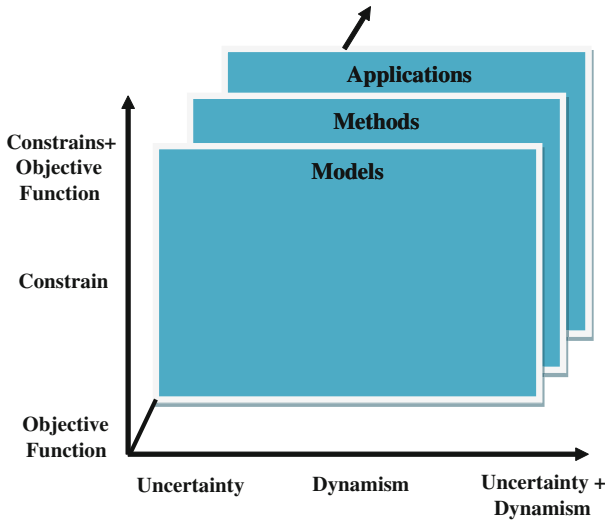


Fig. 1 Dynamic and uncertainty characteristics of a dynamic optimization problems (Cruz et al. 2011)

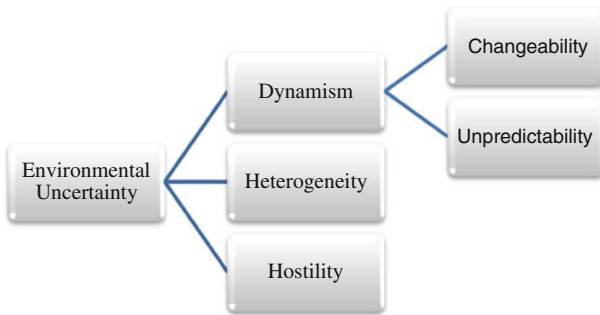


Fig. 2 Type of environmental uncertainty

Different DOPs tried to be solved in the literature for different scopes by using various approaches. Some of the existing studies on DOPs via agent-based modeling propose only frameworks some other studies also present applications. It is remarkable to state that a framework is not a detailed hypothesis or set of hypotheses; rather, it is a suggested point of view for an attack on a scientific problem, often suggesting testable hypotheses (Aggestam and Söderström 2005). Therefore, studies covering solely the frameworks do not provide comparable outputs. Some of the applications cover real-life instances and some have hypothetical data obtained via simulation. Some of the studies attempting to solve DOPs by using agent-based modeling use test problems that have been previously presented in the literature. Thereby, they compare their findings with the best known results in the literature. Some other studies compare the results with themselves by providing different methodologies to solve the problems. Examples of these cases will be discussed in the applications section.

### 3 Solutions for dynamic optimization problems by agent based modeling

Although some problems like scheduling, production planning and travelling salesman have been previously solved as they are static problems, indeed these problems are not totally static problems due to the nature of the real life. Consider an example of scheduling and production planning problems where new machines with advanced capacities are required to be included to the systems or some machines are required to be removed from the system and/or new products are can also be introduced to the system etc. Another example is for a more real life oriented travelling salesman problem (TSP) has been defined by (Homayounfar et al. 2003). They defined three different types of changes for TSP:

- \* Changing the distances (time) between the cities (due traffic congestion, road repair etc.)
- \* Changing the number of the cities (i.e. adding or deleting some cities)
- \* Swapping the cities

A distance (e.g. the distance between the first and second cities) is changed after a specific time or specific generation, determined by Dynamic Frequency. The amount of change is referred to as the Dynamic rate. These two parameters are set prior to the test. After each period of time, which is after a generation specified by Dynamic Frequency, Dynamic rate is added to the specified cost value (i.e. distance between city 1 and 2). This shifts the peak (optimum cost) to another location, so a new global solution can be found (Homayounfar et al. 2003).

Numerous examples can be considered for real-life oriented versions of these well-known problems. The distinction between these conventional problems and real-life oriented ones can be performed as follows. If during the solving of an optimization problem, parameters of the system (i.e. objective and constraint functions) do not change due to nature of the problem then optimization is static, otherwise it is considered to be dynamic (Homayounfar et al. 2003).

In such environments, the optimization process is non-terminating. In the real world, a function to be optimized may vary from time to time and the optima have to be found in time (Guan et al. 2005). Therefore, it is very difficult to optimize such type of dynamic optimization problems with conventional methods which are developed to deal with non-stationary environments. Moreover, it is also quite hard to define dependencies or correlations between the solutions obtained at a time slot and in another time slot. As stated by O'Hare et al. (2007), the real world is both unpredictable and unforgiving. Decisions often need to be made where the contributory evidence is uncertain, incomplete, contradictory and highly dynamic (O'Hare et al. 2007). Therefore, performing design of experiment may not end up with significant findings.

Since finding a feasible solution to the static problem is NP-hard, we must make certain assumptions about the input. Intuitively, if the input is such that even finding a static solution is hard we cannot expect to find a good solution with respect to the dynamic objective function. Thus the problem instance must be "easy enough" that a relatively straightforward agent can find a feasible solution at each time step. In practical terms, this means there must be enough resources to easily satisfy the demand if we ignore the quality of the solution in the sense of the dynamic objective function. In the worst case, we can fall back on the heuristic to find a feasible solution. In fact, agent-based algorithm will degrade gradually to this extreme, but should perform much better in the common case. This challenging dynamism was apparently a significant problem for researchers and it could only be effectively handled by using more advanced approaches like agent-based modeling.



### 3.1 Agents and their features

Although there is no universal agreement in the literature on the precise definition of agent, their property of being autonomous has been common to all definitions (Kulkarni and Tai 2010). An agent is an encapsulated computer system that is situated in some environment and that is capable of flexible, autonomous action in that environment in order to meet its design objectives (Jennings et al. 2001).

Agents that act in an autonomous fashion to perform delegated tasks have aroused much attention over the last decade (Liu et al. 2006) which are named as intelligent agents. An intelligent agent performs interactive tasks tailored to a user's needs without humans or other agents telling it what to do (Máhr et al. 2010). The intelligent agent has been proposed as a software design paradigm, which is different from the sequential, the structural and the object-oriented approaches by its autonomy in deciding when to invoke its action (Parunak 1997).

The fundamental approach of agent based system is to simulate real-world systems with a group of interacting autonomous agents modeled as computer programs (Zhou et al. 2009). Since agent based modeling is a consideration that is developed according to system requirement, it is expected that they have several common and distinguishing characteristics. In fact, in spite of the growing interest in multi-agent systems, there is no agreement on what actually constitutes agenthood, that is, what are the fundamental characteristics of agents (Garcia et al. 2004). On the other hand, Jou and Kao (2002) defined the characteristics of an agent based on agent's behaviors: fundamental, auxiliary, implicit and application oriented (role based). The first one is fundamental characteristics. As a goal is state of affairs to be achieved in the environment, an agent must perceive and process the environmental changes (Jou and Kao 2002). The second one is auxiliary characteristics. An agent also needs information from users and other agents through communication mechanisms such as user interfaces and agent communication language to complete the delegated tasks (Jou and Kao 2002). In addition to fundamental and auxiliary characteristics, some implicit characteristics with presumptive feature exist within an agent (Jou and Kao 2002). Trustworthiness (veracity and benevolence), perceptibility, rationality, persistence, flexibility, and competence are some examples (Jou and Kao 2002). The final characteristics are related to specific applications, e.g., the surfing behaviors of cyberspace, the facial expressions attached to entertainment, and so forth (Jou and Kao 2002). All these features make agent technology an interesting approach for a wide set of applications (Garcia-Montoro et al. 2007).

#### 3.1.1 Number of agent types and agents

An agent based model has a set of agents defined by the model creator. Number of agents can be variable or constant. Their types or functions may also vary or they can be assigned for repetitive tasks. In meta-heuristics based agents, number of agents can directly be proportional (or the same) with the population/colony size. Both "number of agent types and number of agents" should be considered in the design of agent-based modeling for DOPs since they can affect the solution space directly. In his study, Tan (1993), claims that, as the number of agents is increased, state space exponentially increases in terms of the number of agents. A large state space means more state exploration for the model that Tan (1993) considers, and this yields slower learning.



### 3.1.2 Communication type

Multi-agent system, also called ‘self organized system’ is a computational system in which multiple interacting intelligent agents work together to solve difficult problems which may be impossible for an individual agent (Yan et al. 2010). Multi-agent based modeling allows complex natural behavior of various interacting entities to emerge from a set of simple individual rules (Razavi et al. 2011). Communication is the most common means of interaction among intelligent agents. Any observable behavior and its consequences can be interpreted as a form of communication (Mataric 1995). Two different approaches have been defined for communication (Genesereth and Ketchel 1994). Direct communication covers the agents handling their own coordination. Designer of an agent-based system can consider direct communication with respect to problem domain. The cost of communication and availability of obtaining qualified solutions are the factors to be considered for employing direct communication mechanisms in the models.

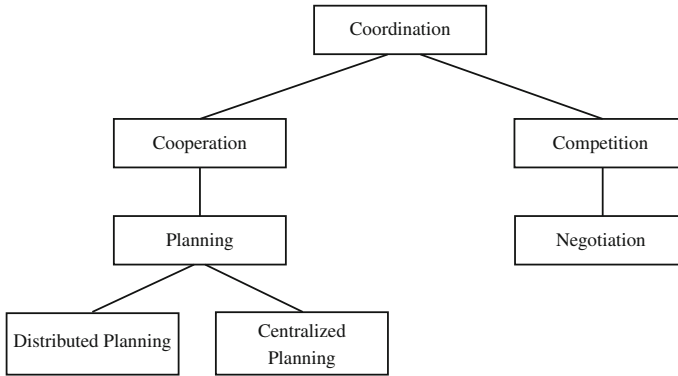
On the other hand, assisted/indirect coordination covers the systems in which agents rely on special system programs to achieve coordination (Genesereth and Ketchel 1994). One of extensively employed indirect communication method is stigmergy. Stigmergy is a class of mechanisms that mediate animal-animal interactions. It consists of indirect communication that is taking place between individuals of an insect society by local modifications induced by these insects on their environment (Hadeli et al. 2004).

Although communication is essential for agent-based modeling, it may not be perfect as expected. In most of the current research on multi-agent systems, people assume that communication of agents is guaranteed (Satoh et al. 2000). However communication can be broken, suspended or delayed depending on the flow of data or information. In this regard, research of problem solving under incomplete communication is very important (Satoh et al. 2000).

### 3.1.3 Coordination type

Since DOPs may have varying objectives, constraints/restrictions and parameters; agents cannot be informed on the global state of the system by themselves. Therefore, in order to achieve global objectives, coordination is essential to allow the agents to adjust their local states or conditions. Communication provides channels for coordination. Coordination is a property of a system of agents performing some activity in a shared environment (Huhns and Stephens 1999). The degree of coordination is the extent to which they avoid extraneous activity by reducing resource contention, avoiding livelock and deadlock, and maintaining applicable safety conditions (Huhns and Stephens 1999). Figure 3 illustrates the taxonomy for the types of coordination defined by (Huhns and Stephens 1999). Panzarasa et al. (2001) prefer to use the generic term “interaction” instead of “coordination”. They (Panzarasa et al. 2001) define cooperation as working together to achieve a common objective and they also define negotiation as coming to a mutually acceptable agreement on some matter (Panzarasa et al. 2001). In other words, cooperation is coordination among nonantagonistic agents, while negotiation is coordination among competitive or simply self-interested agents (Huhns and Stephens 1999).

However, perhaps the most fundamental and powerful mechanism for managing inter-agent dependencies at run time is negotiation—the process by which a group of agents come to a mutually acceptable agreement on some matter (Jennings et al. 2001). Negotiation underpins attempts to cooperate and coordinate (both between artificial and human agents) and is required both when the agents are self interested and when they are cooperative (Jennings et al. 2001).



**Fig. 3** Taxonomy of coordination for agent-based modeling (Huhns and Stephens 1999)

### 3.1.4 Optimization mechanisms

There are different approaches used as optimization mechanism in the agent-based systems for DOPs. These methods can be classified as: heuristic based approaches, exact mathematical methods, and market based approaches. Heuristic based approaches cover, evolutionary based approaches and other techniques like: particle swarm optimization, immune-based algorithms ant colony optimization etc.

*Heuristics:* For large size problems it is hard to obtain the optimal solution due to the large size of the problem files. In order to reduce the computational time, heuristic approaches can be used for obtaining solutions (Baykasoglu et al. 2011). Heuristics includes some set of procedures for obtaining acceptable solutions to the problems by decreasing the time requirements. There is a natural correspondence between autonomous entities and meta-heuristics, and problem solving with an optimization problem (Pelta et al. 2009a,b). In the literature, agents are sometimes defined as individuals, particles etc. (Wang and Shixin 2010) which is the main actors for heuristics. If one takes a look at natural process like bird flocks or fish schools a strong similarity to multi-agent systems can be found very quickly (Wagner et al. 2003). In this regard, there are many studies adapting heuristic approaches to agent-based systems.

*Mathematical methods:* Although it is too complex to deal with DOPs using classical mathematical methods, increasing power of computer technologies let some researchers to use mathematical methods. Change frequency and the response time of the mathematical methods directly affect the solution's availability at the desired time. In this regard, the preferability of mathematical methods appears to be less when compared with the other methodologies.

*Market based approaches:* For an optimization approach to be considered as market oriented, it must at least fulfill the following basic requirements (Karlsson et al. 2007):

- There must be a well-defined market mechanism which includes some notion of prices (which often are expressed in terms of some monetary unit). The market mechanism regulates how negotiations and trade are performed among the participating agents and hence determines how certain commodities can be traded for certain other commodities (Karlsson et al. 2007).
- There must be some arguments for why the agent strategies are reasonably realistic, given the market model. That is, assume we have some model of information available for the

different agents and a well-defined model for how they interact (the market mechanism). Then the strategies must be consistent with the agents' attempt to maximize utility, given bounded rationality (Karlsson et al. 2007).

#### 4 Introduction to agent based dynamic optimization problem solution strategy (ABDOPSS)

The first generic step of developing a solution strategy for any optimization problem is to define specifying class of the problem. Class of the problem is an important factor that is affecting the required solution approach. Therefore, the ingredients that characterize an agent-based modeling approach for a DOP must somehow possess a presentation scheme to provide a solution strategy and a basis for comparison of one with another. Therefore, this paper presents how solution strategies with agent-based approach can be summarized in a concise manner. In this regard, a classification scheme is designed and presented in a tabular form called ABDOPSS (Agent Based Dynamic Optimization Problem Solution Strategy).

ABDOPSS distinguishes different classes of agent based algorithms applied to DOPs (via agent-related features like: communication type, cooperation type and problem related features like: dynamism of objective functions, parameters etc.) by enumerating the fundamental ingredients of each of these algorithms with respect to problem domain. The ingredients that have been considered for ABDOPSS have been discussed in the previous sections.

The main structure of the table consists two rows (row 1, row 2 as illustrated in Table 2), indicating the problem related features (row 1) and agent related features (row 2), respectively. At each corresponding column of a row, an entry, gives the necessary indication for the corresponding criteria. Table 2 shows such an empty table and corresponding values for the table.

#### 5 Sample applications of the ABDOPSS

Since it takes a considerable time to determine features listed by ABDOPSS, 18 articles are exemplified within the contents of this paper. It is remarkable to state here that, this paper does not seek to find all publications on DOPs and represent those using ABDOPSS, but it is rather sought for exemplification of the proposed ABDOPSS. In this regard typical versions of different approaches have been tried to be illustrated in Table 3. Two rows are merged into one for easiness of readability. The studies covered here are roughly classified and some unclear parts made it relatively difficult to prepare ABDOPSS. It is certain that those 18 papers could be better presented by their developers.

Pelta et al. (2009a,b) presented a multi-agent decentralized cooperative strategy (MADCOS) for dynamic optimization problems. In MADCOS, a population of cooperative agents has moved over a grid containing solutions. In order to test and analyze both different communication schemes for the agents and real need of an explicit diversity preserving mechanism in MADCOS, researchers focused on cooperation and diversity mechanisms. Moreover, two types of diversity mechanisms were investigated. Different configurations of the moving peaks benchmark problem were employed as a test bed. A set of computational experiments have been set to evaluate how different communication strategies affect the search; and the usefulness of having explicit and implicit diversity mechanisms. Last offline error (OE) ("accuracy") and relative error between the best found solutions ("error") have been recorded to measure the performance of each configuration. Results showed that having an

**Table 2** Representation of ABDOPSS

Problem related features (row 1)						
Search space/solution environment	Objective function dynamism	Restriction/constraint dynamism	Parameter dynamism	Performance measure	Benchmarked	Applied/framework
Agent related features (row 2)						
Number of agents	Number of agent types	Communication type	Coordination	Coordination type	Population type	Optimization mechanism
If dynamic: D If multiple: M If directly related with population size: PS	Any numeric value	If direct: D If indirect: I	If exists: 1; else: 0	Cooperation, data sharing, decision aid	N/A, static, population, Dynamic population size	If heuristics: H If mathematical methods: MM If market based algorithm: MBA

**Table 3** Examples of ABDOPSS

Authors	Problem related features (row 1)									
	Title	Search space	Objective function dynamism	Constraint dynamism	Parameter dynamism	Performance measure	Benchmarked	Applied/ framework	Real Life/ Test problem/ simulation	
<a href="#">Pelta et al. (2009a,b)</a>	A study on diversity and cooperation in a multi-agent strategy for dynamic optimization problems	D	1	1	1	OE relative error	S	B	TP	
<a href="#">Wang and Liu (2010)</a>	An agent based evolutionary search to dynamic traveling salesman problem	D	0	1	1	Offline performance	OM	B	TP	
<a href="#">Billiau and Ghose (2008)</a>	Robust, flexible multi-agent optimization using SBDO	D	1	1	1	Solution time Avg. # iteration quality	OM	B	TP	
<a href="#">Mähr et al. (2010)</a>	Can agents measure up? A comparative study of an agent-based and on-line optimization approach for a drayage problem with uncertainty	D	0	1	1	Total cost	OM	B	R	

Table 3 continued

Problem related features (row 1)									
Authors	Title	Search space	Objective function dynamism	Constraint dynamism	Parameter dynamism	Performance measure	Benchmarked	Applied/ framework	Real Life/ Test problem/ simulation
Voos (2009)	Agent-based distributed resource allocation in continuous dynamic systems	S	0	I	I	Allocated amounts of steam in the simulation	NA	B	R
Li and Li (2009)	A multi-agent based hybrid framework for international marketing planning under uncertainty	D	0	I	I	-	NA	B	P
Tang et al. (2004)	Wireless-based dynamic optimization for load makeup using auction mechanism	D	0	I	I	Transportation cost dwell time	OM	B	HS
Berro and Duthen (2001)	Search for optimum in dynamic environment: a efficient agent-based method	D	I	I	I	Value of objective function	OM	B	TP

Table 3 continued

Problem related features (row 1)										
Authors	Title	Search space	Objective function dynamism	Constraint dynamism	Parameter dynamism	Performance measure	Benchmarked	Applied/ framework	Real Life/ Test problem/ simulation	
Jiang and Han (2008)	Real time multi-agent decision making by simulated annealing	D	1	1	1	Scalability of SA algorithm, relative payoff comparison	OM	B	HS	
Zhou et al. (2008)	Simulating the generic job shop as a multi-agent system	D	0	1	1	M. Flow time WIP buffer size Resource utilization	OM	B	HS	
González et al. (2010)	A cooperative strategy for solving dynamic optimization problems	D	1	1	1	OE, Mean fitness error	OM	B	TP	
Lepagnot et al. (2010)	A new multi-agent algorithm for dynamic Continuous optimization	D	1	1	1	OE, SD	OM	B	TP	
Yan et al. (2010)	Agent based evolutionary dynamic optimization	D	0	0	1	OE	OM	B	TP	



**Table 3** continued

Problem related features (row 1)									
Authors	Title	Search space	Objective function dynamism	Constraint dynamism	Parameter dynamism	Performance measure	Benchmarked	Applied/framework	Real Life/Test problem/simulation
Hanna and Cagan (2009)	Evolutionary multi-agent systems: an adaptive and dynamic approach to optimization	0/1	0/1	0/1	0/1	-	NA	F	-
Pelta et al. (2009a,b)	Simple control rules in a cooperative system for dynamic optimization problems	D	1	1	1	OE	OM	B	TP
Xiang and Lee (2008)	Ant colony intelligence in multi-agent dynamic manufacturing scheduling	D	0	1	1	M. flow time, M. tardiness, throughput, buffer size machine utilization	S	B	HS
Wang and Usher (2002)	An agent-based approach for flexible routing in dynamic job shop scheduling	D	0	1	1	M. flow time	S	B	HS
Wang et al. (2008)	FBS-enhanced agent-based dynamic scheduling in FMS	D	0	1	1	Avg. value of system objective function Avg. # jobs tardy	S	B	HS (Prototype)

**Table 3** continued

Authors	Agent related features (row 2)						Optimization mechanism
	Number of agents	Number of agent types	Communication type	Coordination	Coordination type	Population type	
Pelta et al. (2009a,b)	M	1	Direct	1	Cooperation	S	H
Wang and Liu (2010)	PS	1	Direct	1	Collaborative	S	H
Billiau and Ghose (2008)	NA	NA	Direct	1	Data sharing	NA	H
Mähr et al. (2010)	M	2	Direct	1	Data sharing	NA	MMH
Voos (2009)	M	3	Direct	1	Data sharing	S	MBA
Li and Li (2009)	M	10	Direct	1	Data sharing	NA	Other
Tang et al. (2004)	M	3	Direct	1	Decision aid	S	H
Berro and Duthen (2001)	PS	1	Direct	0	-	D	H
Jiang and Han (2008)	M	1	Indirect	0	-	S	H
Zhou et al. (2008)	M	6	Direct	1	Data sharing	S	MM
González et al. (2010)	D	1	Direct	1	Centralized	S	H
Lepagnot et al. (2010)	D	2	Indirect	1	Coordination	D	H
Yan et al. (2010)	PS	1	NA	1	Competition	S	H
Hanna and Cagan (2009)	PS	1	NA	1	Data sharing, decision aid	S	H
Pelta et al. (2009a,b)	M	1	Direct	1	Data sharing	S	H
Xiang and Lee (2008)	M	4	Indirect	1	Data sharing	S	H
Wang and Usher (2002)	M	2	Negotiation	1	Contract net (negotiation)	NA	Bidding mechanism
Wang et al. (2008)	M	2	Negotiation	1	Negotiation	NA	H

adaptive probability of cooperation should lead to better results than the ones achieved there. Researchers indicated that regarding diversity, the implicit mechanism provided by their multi-agent model seems to be enough for the scenarios tested. However, it was stated that the explicit mechanism proposed, based on randomization, did not perform as expected.

**Wang and Liu (2010)** presented an agent-based evolutionary search algorithm (AES) for solving dynamic travelling salesman problem (DTSP). In their study, the term agent indicated a candidate position in the search space. Agents resided in a lattice-like environment and a collection of such agents were termed as population. They applied a recombination and local updating procedure to the fittest agent referring to a predefined neighborhood. They also combined the perturbation learning strategy to further reinforce the performance of the local updating rule and hope to seek the global optimum rapidly under changing environments. Dynamic version of KroA100 TSP was selected as the case of implementation. Offline performance which was the best of generation fitness averaged over 100 runs and then averaged over the data gathering period. Experimental result and relevant *t* test result indicated that the performance of AES was excellent. Researchers indicated that the superiority of AES lie in its faster convergence and optimum tracking ability in dynamic environments. It is concluded that AES algorithm had a more quickly and robust convergence capability than standard genetic algorithm (SGA) on dynamic problem.

**Billiau and Ghose (2008)** proposed a new algorithm for solving distributed constrained optimization problems (DCOPs). The proposed algorithm, which has been called as Support Based Distributed Optimization (SBDO), has used agent level objectives instead of weighted or soft constraints that have been used by other DCOP algorithms. In DCOPs, constraints/objectives of the problem change over time by adding or removing constraints/objectives. Researchers compared the performance of this algorithm with asynchronous distributed optimization (ADOPT) and distributed pseudotree optimization procedure (DPOP) by using 120 meeting scheduling problem. Result showed that, although SBDO algorithm has not guaranteed to find the optimal solution, it reliably found solutions within two percent of the optimal solution. Results also revealed that SBDO was able to find near optimal results significantly faster than ADOPT and in a comparable time to DPOP.

**Mähr et al. (2010)** compared two structurally different planning approaches, which were agent based solution and on-line optimization approach, for drayage operations in an uncertain environment. The structurally differentiating feature of the two solution approaches was the level of control: centralized (on-line optimization) and decentralized (agent based solution). Dynamic vehicle routing problem (DVRP) with two types of uncertainty, arrival time and service time, was employed in order to compare the performance of agent-based solution and on-line optimization approach. Moreover, scenarios were developed under different arrival time and service time. In their study, containers and truckers were defined as agent. Each container agent sold itself on an auction to the truck agents. Researchers concluded that when the online optimization approach performed better, it was by capitalizing the optimal (or near optimal) balance between routing and rejection cost. When agents performed better, their flexibility provided by their distributed nature was the competitive advantage.

**Voos (2009)** focused on dynamic resource allocation problems especially in continuous systems. In this study, resource allocation was expressed as an optimization problem, which could be decomposed into single optimization problems, under certain constraints. Researchers proposed to solve this optimization problem in a distributed fashion by using multiple agents. These agents have acted as local optimizer and coordinated their local solutions to an overall consistent solution. In this study, market based interaction mechanism was employed and the agents have calculated and negotiated complete supply and demand trajectories using model based predictions. In order to test the performance of the proposed approach,

researcher employed this approach to three technical examples. Results revealed that this approach could be used to cope with resource allocation in dynamic environments.

Li and Li (2009) proposed a multi-agent coordination and integration method which has been called intercommunication job-sharing hybridization, for solving complex problems. These complex problems could be decomposed into smaller separate sub-problems, with communications/exchange between sub-problems identified, human decision-makers' roles clearly defined and managerial judgment incorporated. With the proposed method, the overall problem was divided into distinct jobs which were then assigned to relatively independent and distributed agents. These agents have shared data, information knowledge and carried out different tasks synchronously or asynchronously in the context of Internet/intranets/extranets to produce solutions. In addition, these agents have linked human participants' judgments and intuition together for joint problem-solving. The architecture of the Internet-enabled multi-agent-based hybrid support system for international marketing planning was exemplified. Researchers constructed a prototype, which has been called Agent International, of the proposed multi-agent hybrid framework. This prototype covered some key features of the conceptual framework and its architecture. The potential and value of the prototype was evaluated by a questionnaire which was prepared and delivered to the corresponding firms in London. According to the responses, researchers concluded that the prototype system was rated somewhat moderately in helping understand pertinent marketing decision making factors, exploring various alternatives, performing analysis, coping with uncertainty, incorporating managers' judgment an improving the confidence and quality of international marketing decision-making.

Tang et al. (2004) presented an auction based dynamic optimization decision support system. The presented decision support system was applied in solving an automobile load makeup planning problem. The proposed system included three types of agents. These were the yard agents, representing the shipping yard, the truck scheduler agent, representing the transportation company who sold trucks and executed transportation and load agent representing a truck during the planning state. Each type of agent had its own behaviors. In order to solve the static load make up problem, researchers implemented 3 heuristics which were empirical (EM), minimum spanning tree (MST), vehicle routing optimization (VRO). In addition to these heuristics, a virtual market enabled by auction mechanism was employed to solve dynamic optimization problem. Researchers developed the mixture of static optimization with MST algorithm and dynamic optimization with the auction mechanism, MST Dyn, at the beginning of the day, they apply static optimization on the guaranteed information, and then they apply the dynamic optimization until the loads are fixed at the beginning of lining up. In order to test the performance of the algorithms, same scenario was used and each algorithm has run for 120 days. Transportation cost and dwell time were selected as the performance measures. Results showed that, EM algorithm produced the worst performance and MST Dyn the best. The MST algorithm produced the worst performance and EM algorithm the best based on the transportation cost. Researchers indicated that MST Dyn algorithm produced a little bit more transportation cost than EM algorithm. Therefore, it was concluded that MST Dyn algorithm produced the best comprehensive performance.

Berro and Duthen (2001) presented an optimization method in dynamic environment. The proposed method, using software agents, tried to provide accurate solutions and react quickly to changes in the state of the problem. In this method, software agents were randomly created and they try to colonize an optimum of the function. The system composed of two parts namely control system and agent. An agent has not communicated but it perceived the presence of other agents. When the functions to optimize and the dimensions of the search space have been defined, the user must specify two parameters namely FORCE which would

influence the searching speed and EPSILON which calculated precision of the optimal points. In order to test the proposed method, 2 cases of optimization problems, multimodal function and multi-objective function, were used. The test results were compared with the Genetic algorithm based approaches. Researchers concluded that the first tests result of the proposed method were satisfactory in particular for the computing speed and precision.

Jiang and Han (2008) presented a simulated annealing (SA) based algorithm to solve real time multi-agent decision making problem. In this study, the SA algorithm was implemented in a centralized version and performed by the agents in parallel, without assuming the availability of communications. Since there was no standard benchmark to evaluate multi-agent decision algorithm, a random generator was used to generate all test sets. The SA algorithm was tested by comparing it, with other algorithms, especially with variable elimination (VE) algorithm with respect to the scalability and relative payoff. Results revealed that, the proposed method was almost optimal with a small fraction of the time that VE took to compute the policy of the same coordination problem. In addition, researchers indicated two main benefits of this approach as follows: the time taken by the algorithm has grown polynomial with the number of agents and the algorithm could report a near-optimal answer at any time. Researchers concluded that, SA was a feasible approach for action selection in large complex cooperative autonomous systems.

Zhou et al. (2008) proposed a model combining discrete event systems and multi agent systems (MAS) to simulate a real time job shop. All entities of the generic job shop were modeled as autonomous agents namely job agent, machine agent, work-center agent, shop floor agent, controller agent and job releaser agent. All agents pursued their own interest with unique functions. All communications in the MAS were realized through message passing. Proportion machines busy (work-center), average number in queue (work-center), maximum number in queue (work-center), average daily throughput (shop floor), average time in system (shop floor), average total time in queues (shop floor), maximal size of work in process (shop floor) were selected as performance indicators. Results of the case study demonstrated the advantage of distributed data collection and analysis. It was indicated that, the case study also validated the proposed system by statistical analysis and comparison to existing simulation results in similar test case.

González et al. (2010) presented a new centralized cooperative strategy based on trajectory methods (tabu/taboo search) for solving DOPs. The proposed strategy was compared with two methods namely a multi-QPSO and Agents. The multi-QPSO is a particle swarm optimization (PSO) variant with multiple swarms and different types of particles where there exists an implicit cooperation within each swarm and competition among different swarms. On the other hand, agents are an explicit decentralized cooperation scheme where multiple agents cooperate to improve a grid of solutions. Researchers tried to assess the possibilities of trajectory methods in the context of DOPs and to draw attention on explicitly including cooperation schemes in methods. A set of solver\threads were consisted in the proposed strategy. Each solver could implement the same of a different resolution algorithm for the problem at hand. The coordinator was responsible for processing the information received from the solver and producing subsequent adjustments of their behavior by sending "orders". Exchange of data was achieved by using a blackboard model, with two blackboards. One of them was written by the solvers that wrote the reports of their performances and read by the coordinator; and another was written by the coordinator that wrote the orders and read by the solvers. The information flow in the proposed strategy was achieved by using the following three steps: firstly, performance information was sent from the solvers to the coordinator, and then this information was processed and stored by the coordinator and finally, the coordinator sent directives to the solvers. A set of rule, based on Reactive Search

Ideas, was employed to control the solvers. In order to test compare the performance of the proposed strategy, moving peaks benchmark problem and three commonly used multimodal real test functions were selected. To emphasize the importance of dynamism and optimal tracking, and to reduce the number of variables to control in the experiments, only the position of the peaks was altered. Proposed strategy and the other two methods were compared according to offline error. Just before a change, offline error of each algorithm was recorded and this value was averaged over all changes, for all independent runs. Therefore, mean fitness error was calculated. Researchers indicated that the proposed strategy could consistently outperform the results of the two other methods. They concluded that the cooperation included in agents provided some benefits over multi-QPSO on the easier problems, but since the optimization done in agents relied only on using simple random perturbations of solutions it may be enough to cope with more difficult problems, even with the help of the cooperation.

Lepagnot et al. (2010) presented a new method called multi agent dynamic optimization (MADO) to solve dynamic optimization problems. In MADO, a population of agent has explored the search space. Three modules namely, memory module, agent manager and coordinator were employed. The number of agents in the system may have varied temporarily, but the number of agents along the whole search process tended to be equal to the predetermined value. This could be achieved by the coordination who would send a delete instruction if the number of agents was higher than a predetermined value. To test the performance of the MADO, moving peaks benchmark problem was employed. Offline error and standard deviation (SD) were selected as performance measure. Different variants of MADO were compared with the proposed one. Results indicated that MADO was better than all the simpler variants. Researchers concluded that the proposed MADO was a promising method for solving dynamic optimization problems.

Yan et al. (2010) proposed an agent based evolutionary search (AES) search method. In AES, a population of agents has represented potential solutions. Similar to EAs, AES gradually converged in the search space during the run, especially when the environment has been stationary for some time. In order to improve the performance of AES for DOPs, two diversity maintaining mechanisms, random immigrants and adaptive dual mapping were employed. Dynamic 0-1 optimization problems which were generated from static optimization problems by using XOR generator were investigated. Nine different dynamic test problems were constructed. The environment was periodically changed every predetermined number of generations. Different change severities were employed to test the performance of AES. The performance of AES was compared with traditional SGA, the primal dual GA and the GA with random immigrant, where the worst 10% individuals were replaced with random individuals every generation. Mean best of generation fitness was selected as performance measurements. According to the mean best of generation fitness, researchers indicated that AES could always outperform other EAs on almost all problems. Researchers stated that the competitive and learning behaviors of agents could always help AES to obtain a better performance than the peer GAs could do. They concluded that some dynamic characteristics of the environments may affect the performance of the algorithm.

Hanna and Cagan (2009) presented an evolutionary multi-agent system (EMAS) for adaptive optimization. EMAS which has employed the evolution of design strategies within a cooperative virtual team has evolved as conditions change and as new solution states were discovered during the optimization process. A set of strategies for creating solutions were represented by population of agents. In EMAS, the strategies for generating solutions have been recombined, altered, and removed by applying genetic operators that are in typical genetic and evolutionary algorithms. However, researchers have made EMAS different from

genetic and evolutionary programming techniques by adding cooperation dimension. Cooperation was employed to combat the problem of not knowing which strategy to use in an unknown but static design space. It has been provided by embodying each strategy in an autonomous agent and allowing the population of agents to communicate. Researchers used the well known combinatorial optimization problem of travelling salesman. Researchers tried to illustrate that the proposed framework was capable of increasing the effectiveness of individual solution strategies by evolving and coordination them a decentralized manner. Three simple heuristic construction algorithms called nearest insertion, farthest insertion and arbitrary insertion were employed. Beside these, three heuristics based algorithms, 2-Opt, 3-Opt and simple mutation were also used. Two basic reduction algorithms, random reduction and best partial reduction were employed. Different from the typical genetic algorithms, gene strings represented agent strategies for generating solutions, not the solutions themselves. The design architecture used in this study was similar to asynchronous team architecture. In this study fitness was based on the ability of the agent to make positive changes in its destination memory. First of all researchers selected a 48-city problem to compare the resulting tours generated by EMAS with those generated by both individual algorithms on their own as well as priori determined hybrid algorithms. When the results of EMAS were compared with the results from running construction algorithms from each starting city and then running improvement algorithms on the resultant tour, researchers found that trials that correlated more strongly with the average patterns had better final values in terms of distance from the optimal value of the average solution quality. In addition to the 532-city problem and 48-city problem, researchers applied these patterns to a team solving a Euclidean 237-city problem modeled on a very large scale integration layout problem. Researchers has seen that the cooperative teams of individual strategies evolved to generate better solutions than both individual strategies alone and a priori set hybrid strategies. It was indicated that the strength of the EMAS algorithm lies in its ability to evolve the best team of agent dynamically. It was also stated that, one of the strength of the EMAS algorithm was as a predictive or learning guide for which set of algorithms or strategies should or should not employed and when. Researchers concluded that, utilizing EMAS in the proposed way has been shown to lower computation time while maintaining or even improving solution quality.

Pelta et al. (2009a,b) investigated if the type of rules employed previously in cooperative systems for static optimization problems have had sense when applied to DOPs. The proposed strategy was based on the joint use of a population of solutions and optimizers (agents). Researchers analyzed the roles that diversity and decentralized cooperation mechanisms in the performance of the methods. Researchers aimed to propose and compare two kinds of control rules to update the solution's set. These rules were a simple frequency based re-sampling (probabilistic) rule and a fuzzy-set based rule. Researchers also tried to understand the behavior of both rules in order to develop more efficient cooperative strategies for DOPs. In this study, cooperation was understood as a mechanism for information sharing. The population of solutions had two purposes which were to serve as an implicit memory that has evolved by means of the action of agents and to be a communication channel for them. An explicit cooperation mechanism was proposed. This mechanism, which was not always triggered, was based on a simple idea. In order to test the performance of the proposed strategy, a set of experiments were developed by using Moving peaks benchmark problem. The main aim of the experiments was to evaluate the behavior of both rules which may trigger the explicit cooperation mechanisms. Offline error was used as performance measurement. The performance of the cooperative system for different setting of each rule and the system's dynamic behavior were also investigated. The results of the proposed strategy were compared with some published results. Researchers stated that the fuzzy-based rule has been better than the



frequency rule. As a conclusion, researchers indicated that both rules have been competitive when compared with a state-of-the art of the algorithm.

Xiang and Lee (2008) proposed an agent-based dynamic scheduling approach which employs ant colony intelligence (ACI) with local agent coordination. The goal of their study was to represent a dynamic manufacturing system through an MAS. They also used ACI to improve the global performance of the system. In the proposed system, entities were modeled as intelligent agents with related knowledge of their own functions and goals. MAS was used to provide parallel execution of commands. Beside this, MAS had the intelligence of negotiation to enhance system performance. The agent coordination mechanism used in the study was inspired by both foraging and division of labor of ant colony in MAS. There were five types of agents in the proposed MAS. They were order agents, job agents, machine agents, work center agents and shop floor agents. Researchers indicated that, different from the previous studies, a more generic manufacturing model with less unrealistic assumptions was considered. Furthermore, ACI was integrated with both machine agents and job agents to solve both task allocation and task scheduling problems. Two types of disturbances were introduced. One of them was resource related disturbance including machine break down and machine recovery. The impact of integrating ACI in agent coordination was investigated by simulating a realistic shop as a multi-agent manufacturing system. In this disturbance, unreality was expressed in terms of mean time between failure and the mean time to repair. Another was source related disturbance including new order\job arrival and existing order\job cancellation. To types of agent coordination, namely coordination based on ACI (MAS + ACI) and coordination using FIFO dispatching rule (MAS + FIFO) were compared. Mean flow time, mean tardiness, throughput, buffer size and machine utilization was employed to measure the performance of the proposed approach. Results showed that a MAS + ACI reduced buffer size, max queue number, mean flow time and tardiness. Researchers concluded that a MAS + ACI were outperformed MAS + FIFO.

Wang and Usher (2002) presented an agent-based job shop model which employed the contract-net protocol as an agent's negotiation mechanism. In this study, two types of agents, named as job agents and machine cell agents, were used with pure hierarchical control structure. Researchers considered routing flexibility to provide more options for job agents. Average flow time and average queue time were selected as performance measurement. In order to measure the impact of the collaborative factor that was incorporated into the contract-net protocol on the performance, a job shop with five different loading levels were simulated. According to the findings, the collaborative factor did not have much effect on mean flow time when the system load is light, but a significant decrease was observed when the system was heavily loaded. When they examined the average queue time for jobs at each machine cell, they observed that the negotiation scheme with the proposed factor reduced the WIP levels of the bottleneck machine cells when the system was under heavy load. Based on the simulation results, researchers concluded that the collaborative factor could improve the performance of the contract net-based negotiation scheme in agent-based scheduling problems.

Wang et al. (2008) proposed a multi-agent approach to study the dynamic scheduling problem in a flexible manufacturing system (FMS) shop floor. The proposed approach was combined with a filtered beam search (FBS). Researchers aimed to show the feasibility of the proposed approach and to evaluate the approach via computational experiments. Dynamisms in this system were provided by new job arrivals. Minimization of a weighted quadratic tardiness function was selected as the objective function. There were two types of agents, a system optimal agent (SOA) and several cell coordinated agents (CCAs). Cooperation and coordination among distributed CCAs and SOA were employed to realize the scheduling

goal. Five modules which were called as communication, cooperation and coordination, FBS-based algorithm for decision making, execution and monitoring, human interface, were used. In addition to these modules, one local knowledge base and one capacity database were also included. FIPA CNIP-based negotiation protocol was selected. FBS was performed by filtering phase and beam selection phase. A prototype system was built to show the practicability of the proposed approach. The performance of the proposed scheduling scheme on the prototype system was compared to two dispatching rule combinations. Two dispatching rules were used for cell selection, named as dispatching by objective function value and dispatching by make-span, one dispatching rule for routing assignment, named as modified shortest processing time in the selected cell and one dispatching rule for determination of starting time of an operation on the selected machine. When the results on the performance of the average number of jobs tardy were investigated, researchers indicated that the quality of the global schedules generated with the proposed scheduling scheme was better than those of two dispatching-rule combinations. In addition, researchers sought the time required for a complete process of scheduling negotiation. They concluded that the proposed scheduling scheme was promising for real world implementation in multiple manufacturing cells of size.

## 6 Concluding remarks

This paper proposes a classification scheme (ABDOPPS: Agent based dynamic optimization problem solution strategy) for agent-based approaches which are employed for solving dynamic optimization problems (DOPs). In this paper 18 typical articles providing agent based solutions to the DOPs are scanned through the literature and represented using the ABDOPPS.

The ABDOPPS is expected to be beneficial to researchers in many ways. Similarities of the features located in ABDOPPS can be used to define classes of solution strategies by their descriptions. In this regard, classes of the problems may orient researchers to focus on certain strategies. Using the dynamism related features of the corresponding DOPs presented in ABDOPPS, unpredictability levels of certain problems can be determined and be used to reclassify problems. These representation forms can also be used to discover the role of presented features and their importance for solution quality.

## References

- Abbas HA, Bacardit J, Butz VM, LLorà X (2004) Online adaptation in learning classifier systems: stream data mining. Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign, IlliGAL Report No. 2004031
- Aggestam L, Söderström E (2005) Managing critical success factors in a B2B Setting. In: Proceedings of the IADIS international conference e-commerce, pp 101–108
- Allmendinger R, Knowles J (2010) Evolutionary optimization on problems subject to changes of variables. In: Schaefer RC, Kolodziej J, Rudolph G (eds) Parallel problem solving from nature-PPSN XI, vol 6239. Springer, Berlin, pp 151–160
- Baykasoglu A, Unutmaz Durmusoglu ZD, Gorkemli L (2011) Solving vehicle deployment planning problem by using agent based simulation modeling. In: Proceedings of 2nd international symposium on computing in science & engineering, Kusadasi, Aydin, Turkey, pp 338–340
- Baykasoglu A, Unutmaz Durmusoglu ZD (2011) Dynamic optimization in a dynamic and unpredictable world. In: Proceedings of Portland international conference on management of technology (PICMET'11), Portland, Oregon, USA, pp 2312–2319

- Berro A, Duthen Y (2001) Search for optimum in dynamic environment: an efficient agent-based method. In: Genetic and evolutionary computation conference. Workshop Program, San Francisco, CA, pp 51–54
- Billiau G, Ghose AK (2008) Robust, flexible multi-agent optimization using SBDO. Decision Systems Lab/Center for Software Engineering, Report, Report No: 2008–TR03
- Borst SC, Buvanewari A, Drabeck LM, Flanagan JM et al (2005) Dynamic optimization in future cellular networks. *Bell Labs Tech J* 10(2):99–119
- Bui LT, Michalewicz Z, Parkinson E, Abello EM (2011) Adaptation in dynamic environments: a case study in mission planning. *IEEE Trans Evol Comput*. doi:[10.1109/TEVC.2010.2104156](https://doi.org/10.1109/TEVC.2010.2104156)
- Calégari P, Coray G, Hertz A, Kobler D, Kuonen P (1999) A taxonomy of evolutionary algorithms in combinatorial optimization. *J Heuristics* 5(2):145–158
- Corchado JM, Glez-Bedia M, De Paz Y, Bajo J, De Paz JF (2008) Replanning mechanism for deliberative agents in dynamic changing environments. *Comput Intell* 24(2):77–107
- Cruz C, González JR, Pelta DA (2010) Optimization in dynamic environments: a survey on problems, methods and measures. *Soft Comput* 15(7):1427–1448
- Cruz C, González JR, Pelta DA (2011) Models of decision and optimization (MODO) research group web. <http://www.dynamic-optimization.org>. Accessed 30 Aug 2011
- Fisher M, Bordini RH, Hirsch B, Torroni P (2007) Computational logics and agents: a road map of current technologies and future trends. *Comput Intell* 23(1):61–91
- García AF, De Lucena CJP, Cowan DD (2004) Agents in object-oriented software engineering. *Softw Pract Exper* 34(5):489–521
- García-Montoro C, Vivancos E, García-Fornes A, Botti V (2007) A software architecture-based taxonomy of agent-oriented programming languages. In *Proceedings of Languages, methodologies and Development tools for multi-agent systems*
- Genesereth MR, Ketchel SP (1994) Software agents *Communication of the ACM* 37(7):48–53
- González JR, Masegosa AD, García (2010) A cooperative strategy for solving dynamic optimization problems. *Memetic Comput* 3(1):3–14
- Guan SU, Chen Q, Mo W (2005) Evolving dynamic multi-objective optimization problems with objective replacement. *Artif Intell Rev* 23:267–293
- Hadeli P, Valckenaers P, Kollingbaum M, Van Brussel H (2004) Multi-agent coordination and control using stigmergy. *Comput Ind* 53(1):75–96
- Hanna L, Cagan J (2009) Evolutionary multi-agent systems: an adaptive and dynamic approach to optimization. *J Mech Des* 131(1):011010-1–011010-8
- Homayounfar H, Areibi S, Wang F (2003) An advanced island based GA for optimization problems. In: *Proceedings of the international DCDIS conference on engineering applications and computations*, pp 46–51
- Huhns MN, Stephens LM (1999) Multi-agent systems and societies of agents. In: Weiss G (ed) *Multi-agent systems*. MIT Press
- Jennings NR, Faratin P, Lomuscio AR, Parsons S, Sierra C, Wooldridge M (2001) Automated negotiation: prospects, methods and challenges. *Group Decis Negot* 10(2):199–215
- Jiang D, Han J (2008) Real time multi-agent decision making by simulated annealing. In: Tan CM (Ed) *Simulating Annealing*. InTechOpen. [http://www.intechopen.com/articles/show/title/real\\_time\\_multiagent\\_decision\\_making\\_by\\_simulated\\_annealing](http://www.intechopen.com/articles/show/title/real_time_multiagent_decision_making_by_simulated_annealing). Accessed 30 July 2011
- Jin Y (2004) A tutorial on evolutionary computation in dynamic and uncertain environments. In: CEC'04, Portland, USA
- Jou SH, Kao SJ (2002) Agent-based infrastructure and an application to internet information gathering. *Knowl Inf Syst* 4(1):80–95
- Jung Y, Kim M, Masoumzadeh A, Joshi JBD (2011) A survey of security issue in multi-agent systems. *Artif Intell Rev* doi:[10.1007/s10462-011-9228-8](https://doi.org/10.1007/s10462-011-9228-8)
- Karlsson M, Ygge F, Andersson A (2007) Market-based approaches to optimization. *Comput Intell* 23(1):92–109
- Kulkarni AJ, Tai K (2010) Probability Collectives: a multi-agent approach for solving combinatorial optimization problems. *Appl Soft Comput* 10(3):759–771
- Lepagnot J, Nakib A, OulHadj H, Siarry P (2010) A new multi-agent algorithm for dynamic continuous optimization. *Int J Appl Metaheuristic Comput* 1(1):16–38
- Li S, Li JZ (2009) A multi-agent-based hybrid framework for international marketing planning under uncertainty. *Int J Intell Syst Acc Financ Manag* 16:231–254
- Liu X, Xu K, Liu H (2006) A multi-agent particle swarm optimization framework with applications. In: *Proceeding of 1st international symposium on pervasive computing and applications*, pp 1–6
- Lung RI, Dumitrescu D (2009) Evolutionary swarm cooperative optimization in dynamic environments. *Nat Comput* 9(1):83–94

- Mataric MJ (1995) Issues and approaches in the design of collective autonomous agents. *Robot Auton Syst* 16(2–4):321–331
- Máhr T, Srouf J, De Weerd M, Zuidwijk R (2010) Can agents measure up? A comparative study of an agent-based and on-line optimization approach for a drayage problem with uncertainty. *Transp Res C- Emer* 18(1):99–119
- Neches R, Fikes R, Finin T, Gruber T et al (1991) Enabling technology for knowledge sharing. *AI Mag* 12(3):36–56
- Newkirk HE, Lederer AL (2006) Incremental and comprehensive strategic information systems planning in an uncertain environment. *IEEE T Eng Manag* 53(3):380–394
- O'Hare GMP, O'Grady MJ, Tynan R, Muldoon C, Kolar HR et al (2007) Embedding intelligent decision making within complex dynamic environments. *Artif Intell Rev* 27:189–201
- Panzarasa P, Jennings NR, Norman TJ (2001) Social mental shaping: modeling the impact of sociality on the mental states of autonomous agents. *Comput Intell* 17(4):738–782
- Parunak HVD (1997) "Go to the ant": engineering principles from natural multi-agent systems. *Ann Oper Res* 75:69–101
- Pelta D, Cruz C, González JR (2009a) A study on diversity and cooperation in a multi-agent strategy for dynamic optimization problems. *Int J Intell Syst* 24(7):844–861
- Pelta D, Cruz C, Verdegay JL (2009b) Simple control rules in a cooperative system for dynamic optimization problems. *Int J Gen Syst* 38:701–717
- Razavi SN, Gaud N, Mozayani N, Koukam A (2011) Multi-agent based simulations using fast multipole method: application to large scale simulations of flocking dynamical systems. *Artif Intell Rev* 35:53–72
- Satoh K, Inoue K, Iwanuma K, Sakama C (2000) Speculative computation by abduction under incomplete communication environments. In: *Proceedings of fourth international conference on multi-agent systems*, pp 263–270
- Tan M (1993) Multi-agent reinforcement learning: independent vs. cooperative agents. In: *Proceedings of the tenth international conference on machine learning*, pp 330–337
- Tang K, Kumara SRT, Yee ST, Tes J (2004) Wireless-based dynamic optimization for load makeup using auction mechanism. In: *Industrial engineering research conference (IERC)*
- Teo TSH, King WR (1997) Integration between business planning and information systems planning: an evolutionary-contingency perspective. *J Manag Inf Syst* 14(1):185–214
- Voos H (2009) Agent-based distributed resource allocation in continuous dynamic systems, *InTechOpen. Multi-agent Systems*
- Wagner S, Affenzeller M, Ibrahim IK (2003) Agent-based problem solving: the ant colonies metaphor. In: *Proceedings of the fifth international conference on information integration and web-based applications & Services*, pp 317–323
- Wang D, Liu S (2010) An agent-based evolutionary search for dynamic travelling salesman problem. In: *Proceeding of 2010 WASE International Conference on Information Engineering (ICIE)*, 1, pp 111–114
- Wang D, Shixin L (2010) An agent-based evolutionary search for dynamic travelling salesman problem. In: *Proceedings of WASE international conference on information engineering*, pp 111–114
- Wang S, Xi L, Zhou B (2008) FBS-enhanced agent-based dynamic scheduling in FMS. *Eng Appl Artif Intell* 21(4):644–657
- Wang YC, Usher JM (2002) An agent-based approach for flexible routing in dynamic job shop scheduling. In: *Proceedings of the 11th industrial engineering research conference*
- Xiang W, Lee HP (2008) Ant colony intelligence in multi-agent dynamic manufacturing scheduling. *Eng Appl Artif Intell* 21(1):73–85
- Yan Y, Yang S, Wang D, Wang D (2010) Agent based evolutionary dynamic optimization. In: Sarker RA, Ray T (eds) *Agent-based evolutionary search*, Chapter 5. Springer, Heidelberg pp 97–116
- Yang S (2007) Explicit memory schemes for evolutionary algorithms in dynamic environments. In: Yang S, Ong YS, Jin Y (eds) *Evolutionary computation in dynamic and uncertain Environments*. Springer, Heidelberg pp 3–28
- Zhou R, Lee HP, Nee AYC (2008) Simulating the generic job shop as a multi-agent system. *Int J Intell Syst Technol Appl* 4:5–33
- Zhou Z, Chan WK, Chow JH (2009) Agent-based simulation of electricity markets: a survey of tools. *Artif Intell Rev* 28(4):305–342