GRID COMPUTING FOR SOLVING THE OPTIMAL WINNER DETERMINATION PROBLEM

Dalila Boughaci, Louiza slaouti and Kahina Achour

University of Sciences and Technology Houari Boumediène
Department of Computer Science
Laboratory of Research and Artificial Intelligence
BP 32 El-Alia, Beb-Ezzouar, 16111, Algiers, Algeria
E-mail dboughaci@usthb.dz, dalila_info@yahoo.fr

Abstract

Grid computing is a recent technology that has become extremely popular to optimize computing resources and manage data and computing workloads. In this paper, we present a parallel genetic algorithm improved by using a stochastic local search for solving the optimal winner determination problem (WDP) in combinatorial auctions. We give also a parallel implementation of the proposed approach on grid by using JSDL job document description.

Keywords - Grid computing, genetic algorithm, stochastic local search, parallelism, winner determination problem, optimization.

1 INTRODUCTION

Grid computing is an innovative approach that has received much attention. It is a powerful way for solving large scale optimization problems that single machines cannot process [4, 5, 6, 7]. In this work, we are interested in the optimal winner determination problem (WDP) in combinatorial auctions. Given a set of bundles bids, the winner determination problem is to decide which of the bids to accept. More precisely, the WDP is finding an allocation that maximizes the auctioneer's revenue, subject to the constraint that each item can be allocated at most once.

This paper tries to propose a parallel genetic algorithm combined with a stochastic local search for the winner determination problem, and implemented on a grid computing platform.

The rest of this paper is organized as follows: The second section gives an overview of both the winner determination problem and the parallel hybrid genetic algorithm for the WDP. The third section proposes an implementation of the proposed approach on a grid platform. Finally, the fourth section concludes and gives some future works.

2 BACKGROUND

The aim of this section is to give a background on the Winner determination problem and the parallel genetic local search method for the WDP.

2.1 Winner determination problem

The winner determination problem (WDP) is a core problem in combinatorial auction (CA). CA is an auction that allocates a set of many goods to bidders in the presence of substitutes and complements. The winner determination problem is a complex problem. It is the problem of finding winning bids that maximize the auctioneers' revenue under the constraint that each good can be allocated to at most one bidder [15, 16].

The WDP can be stated as follows:

Let us consider a set of m items, $M=\{1, 2, \dots m\}$ to be auctioned and a set of n bids, $B=\{B_1, B_2, \dots B_n\}$. A bid B_j is a tuple (S_j, P_j) where S_j is a set of items, and P_j is the price of B_j $(P_j>0)$. Further, consider a matrix $a_{m\times n}$ having m rows and n columns where $a_{ij}=1$ iff the item i belongs to S_j, $a_{ij}=0$, otherwise.

Finally the decision variables are defined as follows: x_j=1 iff the bid B_j is a winning bid and x_j=0 otherwise. The WDP can be modeled as the following integer program [15, 9].

$$\textbf{Maximize} \quad \sum_{j=1}^{n} P_j x_j \qquad (1.1)$$

$$\textbf{Under the constraints}: \sum_{j=1}^{n} A_{ij} x_j \leq 1 \qquad i \in \{1 \ldots m\} \qquad (1.2)$$

$$xj \in \{0, 1\} \qquad (1.3)$$

The objective function (1.1) maximizes the auctioneer's revenue which is equal to the sum of the prices of the winning bids. The constraints (1.2) express the fact that each item can be allocated to at most one bidder. Due to the free disposal assumption, some items could be left uncovered.

The winner determination problem is NP-Complete [12, 8]. Several methods have been studied for the WDP. Among these methods, we can find: the Branch-on-Items (BoI), the Branch on Bids (BoB) [13, 14], the Combinatorial Auctions BoB (CABoB) [15], CAMUS (Combinatorial Auctions Multi-Unit Search) [11], the Hybrid Simulated Annealing SAGII [9], the stochastic local search method [3], the memetic algorithms [2] and the hybrid parallel genetic algorithm [1].

2.2 The Parallel Hybrid Genetic algorithm for the WDP

Genetic algorithms [10] are an evolutionary meta-heuristic that have been used for solving difficult problems. The genetic algorithm operates as follows. From a population of points (parents), the algorithm constructs a new population (children) in combining several parents and applying some random modifications (mutation). The selection phase chooses the best points among parents and children to produce the next population for the next iteration. Usually, the genetic algorithms converges, ie, the population has the tendency to lose its diversity, so it loses its efficacy; it is why the convergence is often used like stop criteria. However, the premature convergence of genetic algorithms is an inherent characteristic that makes them incapable of searching numerous solutions of the problem domain why it is frequent to stop searching after a certain number of generations.
Numerous studies suggest the use of other techniques to improve the performance of the standard genetic algorithms such as: the expanding of the size of the population, the use of sub-populations (Parallel genetic algorithms PGA), and a hybridization of a PGA with SLS (stochastic local search) which can be implemented on a parallel architecture.

The main components of the parallel hybrid genetic algorithm for the WDP are given in following:

A. The individual representation

The individual representation is the same used in [1, 2, 3]. That is, an individual is represented by an integer vector *A* having a variable length bounded by the number of bids *n*, where the components of the vector *A* are the winning bids themselves.

B. The objective function

The objective function called also fitness function or fitness permits to measure the quality of solutions of the WDP. For each individual, a value is associated that measures the quality of such individual in the population. The quality of an individual is given by the sum of the price of winning bids.

C. The selection Operator

The selection step permits to identify the best individuals of the population. Based on the fitness value, a set of high quality individuals is selected to build a new generation.

D. The Crossover Operator

The crossover operator permits to enrich the diversity of the population by manipulating the structure of existing individuals. The proposed crossover operator is planned with two parents and produces one child. The child is produced by concatenating the two parents randomly. We choose randomly an offer from the two parents and add it to the child if it does not cause any conflict with other offers already in the child. We repeated this process until all offers from both parents are examined.

E. The Stochastic Local Search Method

The stochastic local search (SLS) is an improved method used to enhance the quality of the new trial individuals in the current generation. The SLS performs a certain number of local steps that combines diversification and intensification strategies to locate a good solution. The intensification phase consists in selecting a best neighbor solution. The diversification phase consists in selecting a random neighbor solution. The diversification phase is applied with a fixed probability *wp>0* and the intensification phase with a probability *1-wp*. The process is repeated until a certain number of iterations called *maxiter* is reached.

The SLS method is sketched in Algorithm 1.

Algorithm 1: The SLS method.

Require: a WDP instance, an individual *V*, *maxiter*, *wp*
Ensure: an improved individual *V*
for *I* = 1 to *maxiter* **do**
r⇐random number between 0 and 1;
  if *r* < *wp* **then**
  bid = pick a random bid (**Step 1)*
  else
  bid = pick a best bid; (**Step 2)*
  end if
V= *V U* {*bid*};
Based on the conflict graph, remove from *V* any conflicting bids;
end for
return the best individual found.

F. The conflict graph

The conflict graph is a data structure which we have implemented in order to manipulate only feasible solutions. It permits to move from a feasible solution to other ones in the search space. We implemented a conflict graph where the vertices are the bids, and the edges connect the pairs of conflicting bids those sharing items.

G. The migration Operator

The migration process permits to move an individual of a subpopulation to another subpopulation. In this study, the choice of the individual to be moved is based on the fitness criterion where the best individual is selected for migration.

H. The PGASLS for WDP

The hybrid parallel genetic algorithm for the WDP (PGASLS) is a Parallel Genetic Algorithm combined with the Stochastic Local Search method. We used three operators which are the selection, the crossover and the migration operators. The mutation is replaced by the stochastic local search algorithm SLS.

The different steps of the sequential hybrid genetic algorithm are depicted in Figure 1. Figure 2 shows the parallel version of the hybrid genetic algorithm where the population is subdivided into a set of subpopulations. On each subpopulation, we apply the hybrid genetic algorithm process.

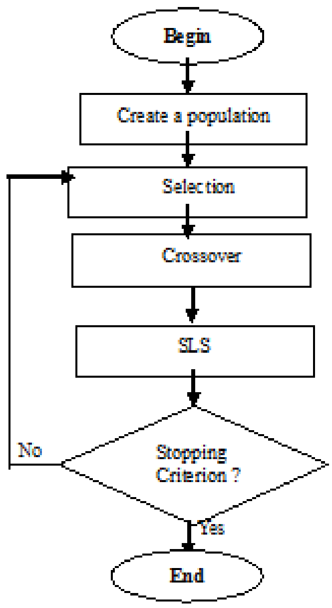


Figure 1. A sequential version of the SGASLS algorithm.

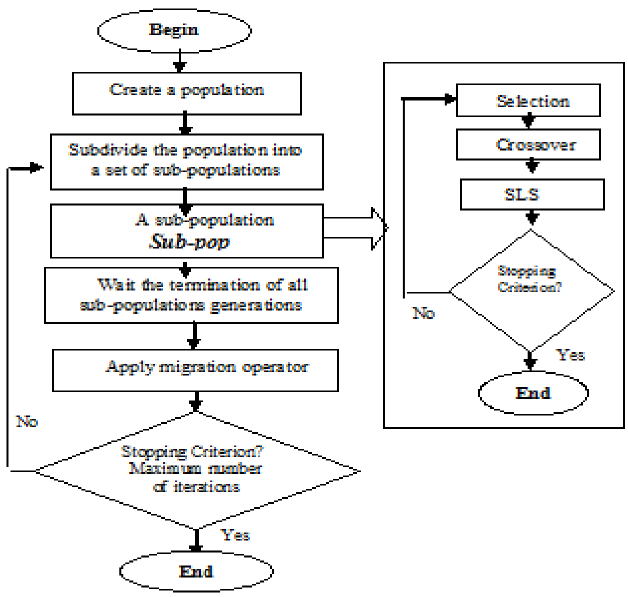


Figure 2. A parallel version of the PGASLS algorithm.

3 PROPOSED IMPLEMENTATION ON GRID

In this section, we explain how we can implement the hybrid genetic algorithm on a grid platform. First, we present the JSDL document used to describe a job to be sent to the grid. Then, we give a model of architecture to launch a set of jobs representing the parallel application of the hybrid genetic algorithm.

3.1 JSDL Document

In order to submit a job to a grid, we need a document that describes this job. This document is in general written with a specific language so-called JSDL (*Job Submission Description Language*, [17].

JSDL defines the structure and the semantic of a job submission in XML format to describe the resources needed for application deployment. JSDL includes a brief description of the application to be deployed as well as the description of the resources that the application process should be launched.

Among the interesting ideas of a JSDL document, we note the possibility to specify whether a part of an application (a process or other) must have exclusive access to a resource. Moreover, this idea could also be useful in the description of resources: it may be that resource can be used by one application at a time, a single process, or a single user, etc..

A major drawback of JSDL description is to mix the resources with the description of the application. The application description is simplified by JSDL (the application is made of independent executables), partial (the structure of the application is not described). Description of resource constraints, in turn, is interesting: JSDL to describe specific requirements imposed by the application of resources. However, the placement of the application resource is optional for JSDL: the program that uses such a document must make choices, find resources, etc.. Finally, JSDL does not specify how, in practice, the process can be run on resources [17].

We give on Figure 3 an example of a JSDL document and the associated XML file is depicted in Figure 4. The two Figures are taken from g-eclipse environment.

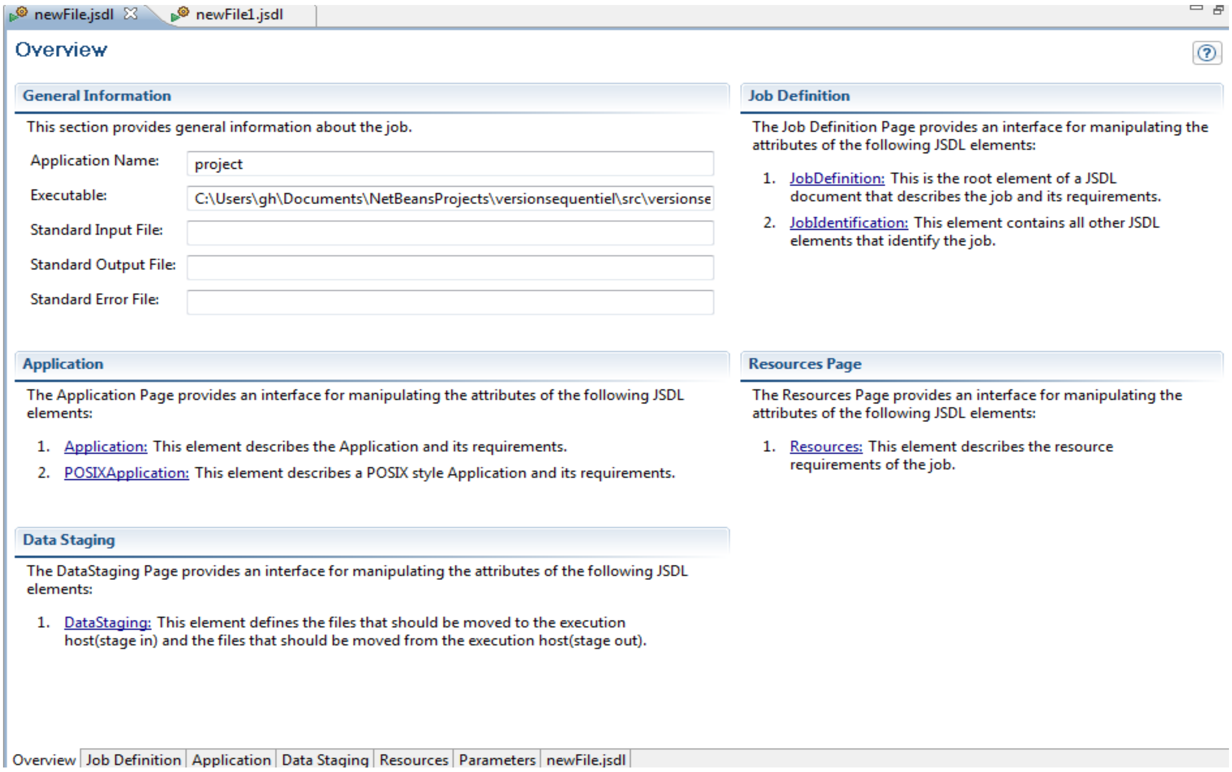
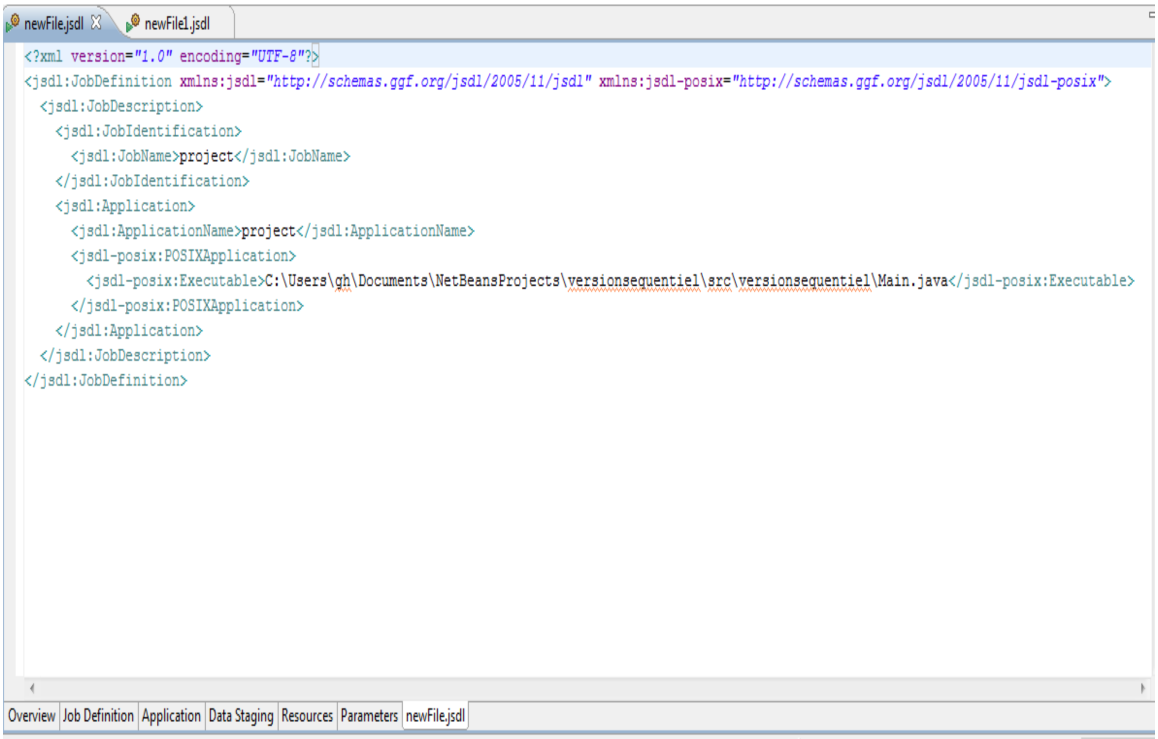


Figure 3 : The JSDL document under G-eclipse [17].



newFile.jsdl newFile1.jsdl

```xml
<?xml version="1.0" encoding="UTF-8"?>
<jsdl:JobDefinition xmlns:jsdl="http://schemas.ggf.org/jsdl/2005/11/jsdl" xmlns:jsdl-posix="http://schemas.ggf.org/jsdl/2005/11/jsdl-posix">
  <jsdl:JobDescription>
    <jsdl:JobIdentification>
      <jsdl:JobName>project</jsdl:JobName>
    </jsdl:JobIdentification>
    <jsdl:Application>
      <jsdl:ApplicationName>project</jsdl:ApplicationName>
      <jsdl-posix:POSIXApplication>
        <jsdl-posix:Executable>C:\Users\gh\Documents\NetBeansProjects\versionsequentiel\src\versionsequentiel\Main.java</jsdl-posix:Executable>
      </jsdl-posix:POSIXApplication>
    </jsdl:Application>
  </jsdl:JobDescription>
</jsdl:JobDefinition>
```

Overview | Job Definition | Application | Data Staging | Resources | Parameters | newFile.jsdl

Figure 4 : The XML file generated by the JSDL under G-eclipse [17].

3.2 The structure of the application on Grid

As already mentioned in section 3.1, the description of an application on the JSDL document is simplistic (the application is made of independent executables), so there is no communication among the jobs. The proposed architecture depicted on Figure 5 is based on this principle, we offer a general architecture of our application and we also give the location of jobs on the grid. As shown on Figure 5, we have a client and several workers. When he received all the results of the different submitted jobs sent by the different workers, the client select the best one to be the solution of the considered problem.

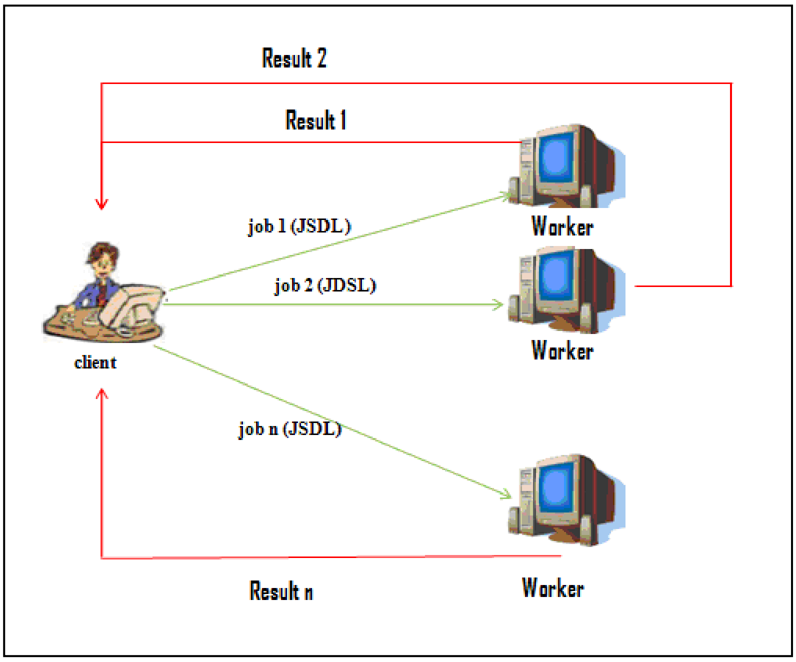


Figure 5. The Proposed architecture.

3.3 Mechanism of job submission on Grid

Figure 6 illustrates the mechanism of the job submission on the grid. The job within the JSDL document will be managed by the resources managers that look for an available resource in the grid to execute the job.

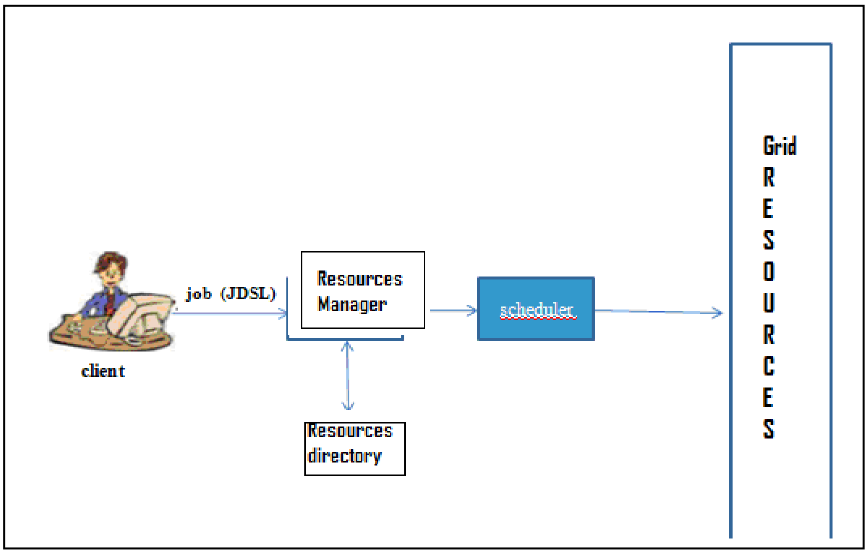


Figure 6. The submission job mechanism on a grid.

3.4 The interactions diagram

We presented in Figure 7, the diagram for the proposed approach for the winner determination approach. The approach is a parallel hybrid genetic algorithm combined with a stochastic local search implemented on a grid. As shown on Figure 7, the concept of migration is eliminated since there is no communication between the JSDL file. .

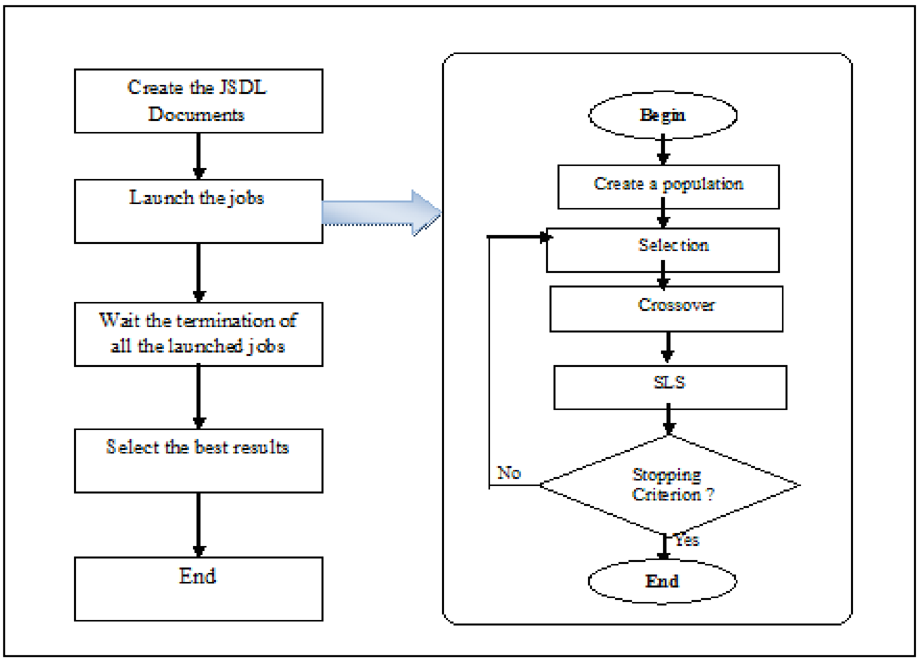


Figure 7. Diagram of the PGASLS for the WDP on grid.

4 CONCLUSION

In this paper, we have detailed the design of our algorithm based on two components: parallel genetic algorithm (GA) and stochastic local search (SLS). The proposed approach is used for solving the winner determination problem in combinatorial auctions. We also gave a parallel implementation of the proposed approach on grid by using JSDL job document description. The numerical results on grid for real benchmarks will be announced in a future work.

5 REFERENCES

[1] Boughaci.D , Slaouti. L, Achour.K: "A Hybrid Parallel Genetic Algorithm for the Winner Determination Problem in Combinatorial Auctions". *In Proceedings of AI-2011, the Thirty-first SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence, pp* : 65-78, *Cambridge, UK*, December 2011. Springer 2011.

[2] Boughaci. D, Benhamou. B, Drias. H: "A memetic algorithm for the optimal winner determination problem", in Soft Computing - A Fusion of Foundations, Methodologies and Applications, Springer Berlin / Heidelberg, vol 13, numbers 8-9, pp: 905- 917, 2009.

[3] Boughaci. D, Benhamou. B, Drias. H: "Stochastic Local Search for the Optimal Winner Determination Problem in Combinatorial Auctions". CP 2008: LNCS pp: 593-597 (September 2008).

[4] Foster, I., Kesselman, C., Nick, J., Tuecke, S.: The Physiology of the Grid: an Open Grid Services Architecture for Distributed Systems Integration. Technical report, Global Grid Forum (2002) 6. National Center for Biotechnology Information, http://www.ncbi.nlm.nih.gov

[5] IImade, H., Morishita, R., Ono, I., Ono, N., Okamoto, M.: A grid-oriented genetic algorithm framework for bioinformatics. New Gen. Comput. 22(2), 177–186 (2004).

[6] Munawar, A.,Wahib,M.,Munetomo, M., Akama, K.: Optimization Problem Solving Framework Employing GAs with Linkage Identification over a Grid Environment. In: CEC 2007: Proceedings of IEEE congress on Evolutionary Computation, Singapore (2007).

[7] Wahib, M., Munawar, A., Munetomo, M., Akama, K.: A General Service-Oriented Grid Computing Framework For Global Optimization Problem Solving. In: SCC 2008: Proceedings of the 2008 IEEE International Conference on Services Computing, Honolulu, Hawaii, USA. IEEE, Los Alamitos (2008).

[8] Fujishima, Y., Leyton-Brown, K., Shoham, Y.: Taming the computational complexity of combinatorial auctions: optimal and approximate approaches. In: Sixteenth International Joint Conference on Artificial Intelligence, pp. 48–53 (1999)

[9] Guo, Y., Lim, A., Rodrigues, B., Zhu, Y.: Heuristics for a bidding problem. Comput. Oper. Res. 33(8), 2179–2188 (2006).

[10] Goldberg. D. E,."Genetic Algorithms in search, Optimization and Machine Learning". *In Wokingham, Addison-Wesley.* 1989.

[11] Leyton-Brown, K., Tennenholtz, M., Shoham, Y.: An algorithm for multi-unit combinatorial auctions. In: Proceedings of the 17th National Conference on Artificial Intelligence, Austin, Games-2000, Bilbao, and ISMP-2000, Atlanta (2000).

[12] Rothkopf, M.H., Pekee, A., Ronald, M.: Computationally manageable combinatorial auctions. Manage. Sci. 44(8), 1131–1147 (1998)

[13] Sandholm, T., Suri, S.: Improved optimal algorithm for combinatorial auctions and generalizations. In: Proceedings of the 17th National Conference on Artificial Intelligence, pp. 90–97 (2000).

[14] Sandholm, T., Suri, S., Gilpin, A., Levine, D.: CABoB: a fast optimal algorithm for combinatorial auctions. In: Proceedings of the International Joint Conferences on Artificial Intelligence, pp. 1102–1108 (2001).

[15] Sandholm, T.: Optimal winner determination algorithms. In: Cramton, P., et al. (ed.), Combinatorial Auctions. MIT, Cambridge (2006).

[16] Vries de S and R. Vohra, 'Combinatorial auctions a survey', In INFORMS Journal of Computing, Vol 15, pp. 284-309, (2003).

[17] g-Eclipse - Access the power of the Grid. http://www.geclipse.org/