A hybrid bacterial foraging optimization method for the permutation flow shop problem

P. Palominos[1] M. Palma[1] V. Parada[2] and L. Quezada[1]

1. Industrial Engineering Department, University of Santiago of Chile
Av. Ecuador 3769, Estación Central, Santiago, Chile
pedro.palominos@usach.cl

2. Informatics Engineering Department, University of Santiago of Chile
Av. Ecuador 3659, Estación Central, Santiago, Chile
victor.parada@usachl.cl

Keywords : Metaheuristics, BFO, SA, permutation flow shop problem.

Abstract

The permutation flow-shop problem (PFSP) represents a particular case of the flow-shop scheduling problem. The goal of permutation flow-shop is to determine an optimal schedule for N jobs on M machines. A job consists of m operations, and the i^{th} operation of each job must be processed on machine j. A job can start on machine j after it is completed on machine j-1; hence, the operating sequences of the jobs are the same on every machine. Each operation has a known processing time p_{ij}, and the objective is to minimize the makespan. Solving the problem requires determining the permutation of jobs that results in the smallest makespan value. This problem belongs to the NP-hard class [1].

The study of the PFSP has been strongly influenced by Johnson's early work [2], in which the analysis of the problem for two machines seemed to capture the nature of bigger problems. Because the algorithm proposed by Johnson always generates an optimal solution for two machines, it provided a basis for the development of heuristic approaches to bigger problems, such as those of J. N. Gupta [3] and Campbell *et al.*[4]. In problems of three or more machines, Johnson's algorithm can be applied as a first approximation by grouping machines into two "virtuals." However, this type of heuristic becomes inefficient when the number of machines is greater than three. For this reason, constructive heuristics have been generated, but at a high computational cost [5]. Alternatively, some studies have used the branch and bound algorithm to find optimal solutions [6], but solving the problem by mathematical programming has not been very efficient for large or even middle-sized problems.

Among several metaheuristics that have emerged in recent decades [7] to tackle difficult problems such as PFSP, the bacterial foraging optimization method (BFOM) has shown interesting results. New research suggests that microbial life is even richer than once believed: highly social, closely connected through networks, and full of interactions. Bacteria communicate through molecules that are similar to pheromones, as occurs in ant colonies [8] [9][10]. In this cell network, microbes are capable of collectively reacting to changes in their surroundings, for example, conspiring against their own species, building mutually beneficial alliances with other types of bacteria, gaining advantages over their competitors, and communicating with their hosts. This behavior has been emulated to give rise to BFOM, which presents promising results for continuous optimization problems [9] [10]. BFOM has also been applied successfully in the tuning of adaptive controllers [10], in the tuning of PID controllers [11], and in portfolio optimization with liquidity risk [12]. Furthermore, it has been used as a support tool for improving the performance of other metaheuristics such as genetic algorithms [11], Harmony Search [13], and ant colony optimization [14]. In all cases, the applications were made over continuous functions, unlike the discrete objective function and set of constraints that define the PFSP. Although, the optimization ability of BFOM has been questioned when compared with other classic algorithms [15], one can expect a better computational performance in discrete optimization problems because its flexibility allows combining different search techniques into a single framework. Recent studies show that the automatic merging of different search techniques provides a powerful tool for tackling combinatorial optimization problems [16]. Thus, BFOM could lead to a new flexible method to be used with other combinatorial optimization problem.

This paper proposes an adaptation of BFOM to the PFSP. The original algorithm that was designed and tested for continuous optimization problems is extended for this typical case of the family of scheduling problems. BFOM is also hybridized with the concept of acceptance probability of a solution during the search process such as in the simulated annealing metaheuristic. The resulting algorithm is a cyclic combination of different search techniques. The hybrid metaheuristic was evaluated with the OR-Library data set that contains the most challenging problem instances. The solutions found have an average relative percentage deviation with respect to the optimal solution of 0.53% outperforming the results obtained with other 13 heuristics and metaheuristics already existing in 8 of the 12 subsets studied. Our metaheuristic is effective at solving the Permutation Flow Shop Problem and provides a new approach that may be applied to other NP-hard problems.

References

[1] M. Garey and D. Johnson (2003). *Computers and intractability: a guide to the theory of NP - completeness*. New York: W.H. Freeman and Co.

[2] S. Johnson (1954). Optimal two- and three-stage production schedules with setup times included. *Naval Research Logistics Quarterly*, *1*(1), 61–68.

[3] N. Gupta (1971). A functional heuristic algorithm for the flowshop scheduling problem. *Operational Research Quarterly (1970-1977)*, *22*(1), 39-47.

[4] H. Campbell, R. Dudek, and M. Smith (1970). A heuristic algorithm for the *n* job, *m* machine sequencing problem. *Management Science*, *16*(10), B630-B637.

[5] A. Agarwal, S. Colak and E. Eryarsoy (2006). Improvement heuristic for the flow-shop scheduling problem: An adaptive-learning approach. *European Journal of Operational Research*, *169*(3), 801-815.

[6] R. Companys, R. (1999). Note on an improved branch-and-bound algorithm to solve n/m/P/Fmax problems. *Top*, *7*(1), 25-31.

[7] G. Talbi (2009). *Metaheuristics: from design to implementation*. John Wiley & Sons.

[8] B. Bassler (2002). Small talk. Cell-to-cell communication in bacteria. *Cell*, *109*(4), 421-424.

[9] Y. Liu and K. Passino, (2002). Biomimicry of social foraging bacteria for distributed optimization: models, principles, and emergent behaviors. *Journal of Optimization Theory and Applications*, *115*(3), 603-628.

[10] K. Passino(2002). Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Systems*, *22*(3), 52-67.

[11] D. Kim, A. Abraham and J. Cho (2007). A hybrid genetic algorithm and bacterial foraging approach for global optimization. *Information Sciences*, *177*(18), 3918-3937.

[12] B. Niu, Y. Fan, H. Xiao and B. Xue (2012). Bacterial Foraging Based Approaches to Portfolio Optimization with Liquidity Risk. *Neurocomput.*, *98*, 90–100.

[13] B. Shivakumar and T. Amudha (2012). A hybrid bacterial swarming methodology for job shop scheduling environment. *Global Journal of Computer Science and Technology*, *12*(10-A).

[14] S. Narendhar and T. Amudha,(2012). A Hybrid Bacterial Foraging Algorithm For Solving Job Shop Scheduling Problems. *arXiv:1211.4971 [cs]*. Retrieved from http://arxiv.org/abs/1211.4971.

[15] X. Yan, Y. Zhu, H. Zhang, H. Chen and B. Niu (2012). An Adaptive Bacterial Foraging Optimization Algorithm with Lifecycle and Social Learning. *Discrete Dynamics in Nature and Society, 2012*.

[16] E. Burke, M. Gendreau, M. Hyde, G. Kendall, G., Ochoa, E. Özcan and R. Qu, R. (2013). Hyper-heuristics: a survey of the state of the art. *Journal of the Operational Research Society*, *64*(12), 1695-1724.