ISSN 1392 – 124X INFORMATION TECHNOLOGY AND CONTROL, 2011, Vol. 40, No. 2

COMPARISON OF TWO HEURISTIC APPROACHES FOR SOLVING THE PRODUCTION SCHEDULING PROBLEM

Arūnas Andziulis[1], Dalė Dzemydienė[2], Raimundas Steponavičius[3], Sergej Jakovlev[1]

[1] *Klaipeda University, Bijunu Str. 17, LT - 91225, Klaipeda, Lithuania*
[2] *Mykolas Romeris University, Ateities Str. 20, LT - 08303 Vilnius, Lithuania*
[3] *Newcastle University, NE17RU, Newcastle upon Tyne, United Kingdom*
e-mail: arunas@ik.ku.lt; daledz@mruni.eu, s.jakovlev.86@gmail.com

crossref http://dx.doi.org/10.5755/j01.itc.40.2.426

Abstract. Production scheduling problems attract a lot of attention among applied scientists and practitioners working in the field of combinatorial optimization and optimization software development since they are encountered in many different manufacturing processes and thus effective solutions to them offer great benefits. In this work, two commonly used heuristic methods for solving production scheduling problems, namely, the Nearest Neighbor (NN) and Ant Colony Optimization (ACO) have been tested on a specific real-life problem and the results discussed. The problem belongs to the class of Asymmetric Travelling Salesman Problems (ATSP), which is known as a hard type problem with no effective solutions for large scale problems available yet. The performances of the Nearest Neighbor algorithm and the Ant Colony Optimization technique were evaluated and compared using two criteria, namely: the minimum value of the objective function achieved and the CPU time it took to find it (including the statistical confidence limits). The conclusions drawn suggest that on one hand the ACO algorithm works better than NN if looking at the achieved minimum values of the objective function. On the other hand, the computational time of the ACO algorithm is slightly longer.

Keywords: theory of algorithms, production scheduling, asymmetric travelling salesman problem, Ant Colony optimization, nearest neighbor.

1. Introduction

In spite of various methods and techniques being actively and continuously developed for solving different combinatorial optimization problems such as production scheduling [1, 7, 8] this is still an open-end problem in most practical situations. Such methods and techniques can deliver substantial benefits by improving productivity, utilization of resources and time constraint management at different levels of decision-making and manufacturing processes [1, 3, 17, 24]. That is why different types of job shop scheduling and resource allocation problems are becoming an intensively studied field, as they are faced in many industrial areas.

Process scheduling can pose extremely complex combinatorial optimization problems that belong to the NP-hard family. Many research works were devoted for solving the process scheduling optimization problems in different application areas [19, 5, 6, 25, 26, 27]. The problem of finding the best production sequence is generally formulated as the Travelling Salesman Problem [3, 15, 22]. There are generally two ways of solving such problems: exact and heuristic. According to Yagmahan and Yenisey's reported results the heuristic Ant Colony Optimization (ACO) algorithm can be quite effective in solving such job shop scheduling type problems [4]. In recent years, among the various approaches for solving different scheduling problems, there has been also an increasing interest in applying Genetic Algorithms (GA) to solve the combinatorial optimization problems including production process scheduling [2, 10, 20], where Tavakkoli-Moghaddam *et al*. successfully applied genetic algorithm (GA) to solve the quay crane (QC) control and assignment problem, in a container port (terminal) using a mixed-integer programming (MIP) model [16].

Recently, Boland *et al*. addressed the problem of the open pit mining scheduling [18]. They proposed an iterative disaggregation method to solve the problem formulated as a mixed integer program (MIP). Klemmt *et al*. analyzed a hybrid approach for solving scheduling problems [1]. Georgiadis *et al*. presented the development and implementation of a production scheduling system for an electrical appliance manufacturer [17]. Nonas and Olsen proposed a mixed integer linear programming formulation for the scheduling problem together with a set of heuristic strategies [21].

Integer programming models have been widely used for solving different combinatorial optimization tasks [13, 14]. However, the use of exact methods is limited for solving large scale and complex problems hence they are often not applicable in many practical situations, in particular in make-to-order manufacturing [12] where the performance is evaluated by qualitative as well as quantitative information.

The aim of this study was to compare two heuristic approaches, the NN and the ACO, in solving specific actual ATSP problem. The efficiency of the algorithms was measured according to the value of the objective function and the CPU time.

The results of this work can aid in solving similar types of practical problems in the future.

2. Definition of the production scheduling problem

In today's competitive industrial environment the difference between using quickly gained empiric methods and specially designed algorithms for production scheduling can determine whether or not a manufacturing company has a future, because productivity and optimal usage of resources strongly depend on job scheduling which therefore has a major impact on overall effectiveness of the production processes.

During the collaboration with Lithuanian largest candle manufacturing company UAB "Geralda" (Ltd.), it became clear that the main obstacle in further successful development lied in using the right (optimal) production scheduling. The objective was to minimize the job change over times, which in turn would give the highest productivity of the production lines i.e. the minimum makespan. Before presenting the objective function, some technical definitions are as follows: the TSP is defined on a graph $G=(V, A)$ for each separate production line, where V is the set of n (n=48 for each production line) jobs (vertices) and A is the set of change over times (distances). Let $C=(c_{ij})$ be a distance matrix associated with A and $B=(b_y)$ be a job matrix associated with V.

The matrix C is said to be symmetric when $c_{ij} \to c_{ji} = c_{ji} \to c_{ij}$, $\forall (i, j) \in A$ and asymmetric when $c_{ij} \to c_{ji} \neq c_{ji} \to c_{ij}$. If $c_{ij} + c_{jk} \geq c_{ik}$, $\forall i, j, k \in A$, C is said to satisfy the triangle inequality. An assignment based double-index integer formulation $x_{ij} = \{1,0\}$ is used to define the binary variables {1} or {0} used in the description of the objective function where variable {1} is assigned if the distance (i, j) has been used and {0} otherwise. The problem is described as an asymmetric TSP (ATSP) and formulated as follows (1.1):

$$\min \sum_{i=1}^{n-1} \sum_{j=1}^{n-1} c_{ij} x_{ij} , \qquad (1.1)$$

Subject to:

$$\sum_{i=1}^{n-1} x_{ij} = 1, \quad j = 1,...,n, \qquad (1.2)$$

$$\sum_{j=1}^{n-1} x_{ij} = 1, \quad i = 1,...,n, \qquad (1.3)$$

+ sub tour elimination constraints, (1.4)

$$x_{ij} \in \{1,0\}, \ \forall (i, j) \in A, \qquad (1.5)$$

where (1.2), (1.3) and (1.5) are the usual assignment constraints. Constraints (1.4) are used to prevent sub tours, which are degenerate tours that are formed between intermediate jobs and not connected to the origin [23]. These constraints are named as sub tour elimination constraints (SECs).

$$\sum_{i \notin S} \sum_{j \in S} x_{ij} \geq 1, \ \ \forall S \subseteq V \setminus \{1\}, \ S \neq Ø. \qquad (1.6)$$

Constraint (1.6) impose connectivity requirement for the solution, i.e. prevent the formation of sub tours of cardinality S not including the departure jobs.

3. Formulation of the algorithm

In this section, the Nearest Neighbor (NN) algorithm and the Ant Colony Optimization (ACO) algorithm are defined.

3.1. NN heuristic algorithm for solving production control problem

The nearest neighbour (NN) algorithm is a very fast and simple heuristic solution method for production scheduling. Though, Gutin *et al*. showed that NN algorithm, while producing comparatively good solutions with TSPs, may yield poor results with Asymmetric TSPs (ATSPs) [11].

The NN algorithm starts with an arbitrarily chosen job b_1 as partial tour. Then it repeats the following step for $g = 1,...,n-1$: If the current partial tour is $b_1,...,b_g$, then let b_{g+1} be the job closest to b_g (having the smallest c_{ij} value) subject to the condition that b_{g+1} is not already contained in the partial tour; ties are broken arbitrarily.

The sequence of the selected jobs and the sum of distances are the outputs of the algorithm. That way, it could be suggested that it is possible to get a near optimum objective function value with the given NN algorithm.

3.2. ACO meta-heuristic algorithm for solving the production scheduling problem

An Ant Colony Optimization (ACO) technique was used as a paradigm for designing a meta-heuristic algorithm to solve the given production scheduling problem. A conventional ant colony optimization system framework was used, as described in [4], where a set of m artificial ants construct solutions from elements of a finite set of available solution components

$C=(c_{ij})$. A solution construction starts from an empty partial solution $f^p \neq \emptyset$. At each construction step, the partial solution f^p is extended by adding a feasible solution component from the set $N(f^p) \subseteq C$ of components that can be added to the current partial solution f^p without violating any of the defined ACO model constraints. Here the choice of a solution component from $N(f^p)$ is guided by a stochastic mechanism, which is biased by the pheromone associated with each of the elements of $N(f^p)$. When ant *h* has selected job *i* and constructed the partial solution f^p, the probability of selecting job *j* is given by:

$$p_{ij}^h = \frac{\tau_{ij}^{\alpha} \cdot \phi\left(c_{ij}\right)^{\psi}}{\sum\limits_{c_{il} \in N(f^p)} \tau_{il}^{\alpha} \cdot \phi\left(c_{il}\right)^{\psi}}, \; c_{ij} \in N(f^p), \qquad (2.1)$$

where $N(f^p)$ is the set of feasible components; that is, distances (*i*, *l*) where *l* is a job not yet selected by the ant *h*. The parameters α and ψ control the relative importance of the pheromone versus the heuristic information $\varphi(c_{ij})$ which is given by $\varphi(c_{ij}) = \frac{1}{c_{ij}}$. At each iteration the pheromone values are updated by *all* the *m* ants that have built a solution in the iteration itself. The pheromone τ_{ij}, associated with the distance between joining jobs *i* and *j*, is updated as follows:

$$\tau_{ij} = (1-e) \cdot \tau_{ij} + \sum_{h=1}^{m} \Delta \tau_{ij}^{h}, \qquad (2.2)$$

where $e = 0.1$ is the evaporation rate and $\Delta\tau_{ij}^h$ is the quantity of pheromone laid on distance *(i, j)* by ant *h*.

4. Computational results

Computational results show that ACO algorithm on average gave better values of the objective function in comparison with NN algorithm for all production lines, see Table 1.

Table 1. Main computational results

Production line	Method used	CPU time (s)	Value of objective fun., (min)	Mean value, (min)
1	NN	0.1162	540	540
	ACO	0.7251	316	328
2	NN	0.1712	298	298
	ACO	1.6204	256	296
3	NN	0.1644	461	461
	ACO	1.7855	245	269

However, it is interesting to notice that, for the second line the value of the objective function achieved with NN is very close to the mean result of ACO indicating that in some cases NN can be a relevant choice.

The distribution of the values of the objective function obtained using ACO with 18 ants (300 iterations) is illustrated in Figure 1. As one can notice, the distribution of values is not strictly normal, it more resembles a log normal distribution. The percentage of better solutions grew with increasing number of ants in the system.

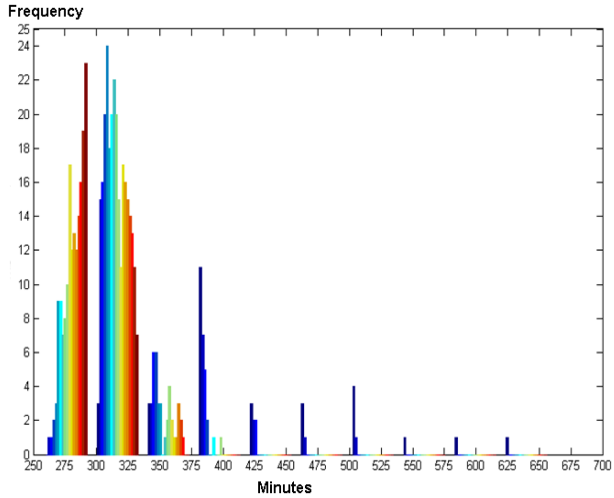


Figure 1. Histogram of values of the objective function obtained using ACO with 18 ants, 300 iterations (2nd production line)

Results also indicate that the ACO algorithm with a small number of ants (up to 16) on average gave worse values of the objective function in comparison with NN algorithm, see Figure 2 (a), and better values using more than 17 ants, see Figure 2 (b).

An important result is that with the fixed amount of ants in the ant colony system increasing the number of iterations does not necessarily lead to a better solution as one can see in Figure 3 (a) and (b).

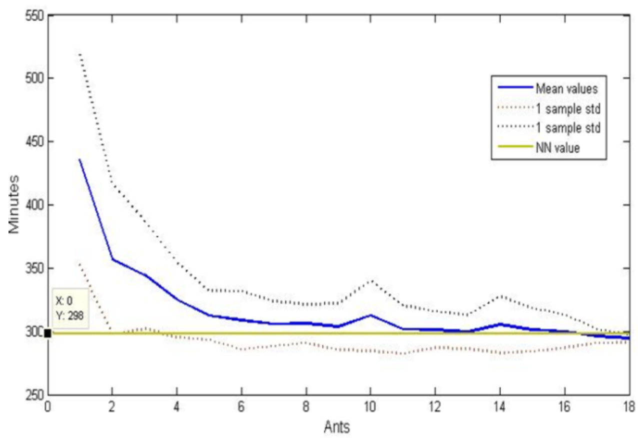


Figure 2 (a). Illustration of results after applying the ACO algorithm with a small number of ants (up to 16)

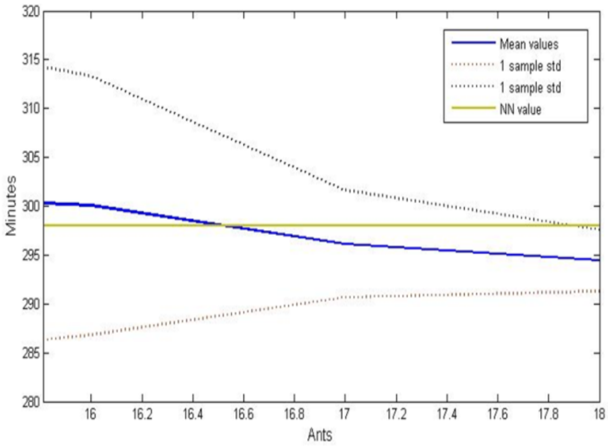
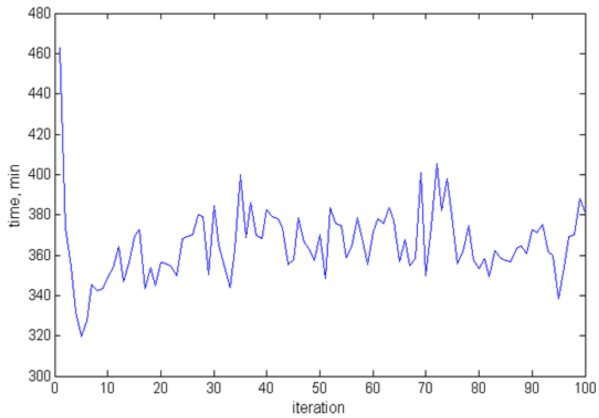
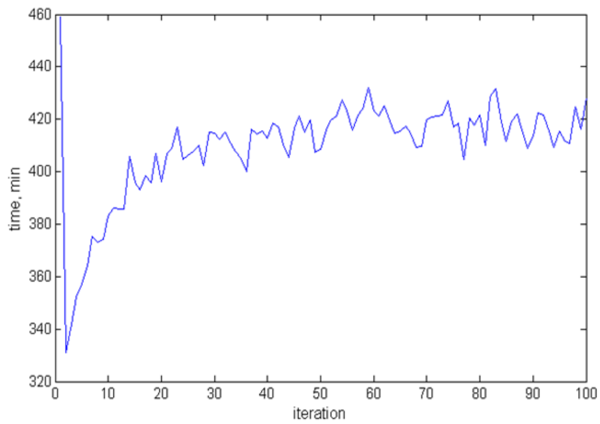


Figure 2 (b). ACO computational results using more than 17 ants (2nd production line)



a)



b)

Figure 3. ACO computational data (average of total distance versus number if iterations) from the second production line using: a) 18 ants; b) 306 ants

After some optimal number of iterations, at which the minimum mean value of the objective function is achieved, the mean value starts increasing with the increasing number of iterations. This suggests that there is no point running a lot of iterations for achieving good results. It was found from the computational results that the optimum range of iterations is around from 5 to 100 (i.e. where the minimum is), see Figure 3 (a), (b).

The reason why the value of the objective function increases with the number of iterations is because trails get too "contaminated" with the pheromone.

5. Conclusions and future work

The NN algorithm showed similar results as in [11] with the first and the third production lines, performing there considerably worse than the ACO algorithm, indicating that the ACO is on average more efficient than NN with respect to the obtained objective function values in similar cases, see Table 1. It is important to note however that for the second production line the results were comparable. With respect to CPU time, NN is significantly quicker. However, in cases when the computational time is not of high relevance as it was in this case the choice of ACO would be more rational.

As Mokotoff and Chretienne [9] suggested, specially developed combinatorial optimization algorithms for production scheduling work better than general methods, thus they can greatly outperform other empirical methods currently very common among production practitioners.

In the future, problem-specific simplifications (adjustments) in the formulation of the ACO algorithm will be implemented to get faster and better results without eliminating critical components for better production scheduling in real situations.

Acknowledgment

The authors would like to thank developers of the project LLII-061 "Development of Joint Research and Training Center in High Technology Area" (Latvia-Lithuania Cross Border Cooperation Programme Under European Territorial Cooperation Objective 2007-2013 Subsidy Contract No: LV-LT/1.1./LLII-061/2010/) for the financial support of these research works.

The authors would also like to thank UAB Geralda (Ltd.) CEO Gintautas Igaris for the materials and additional support while writing the manuscript.

References

[1] **A. Klemmt, S. Horn, G. Weigert, K.J. Wolter**. Simulation-based optimization vs. mathematical programming: A hybrid approach for optimizing scheduling problems. *Robotics and Computer-Integrated Manufacturing*, 2009, *Vol*. 25(6), 917 – 925.

[2] **A. Misevičius, D. Rubliauskas, V. Barkauskas**. Some Further Experiments with the Genetic Algorithm for the Quadratic Assignment problem. *Information Technology and Control*, 2009, *Vol*. 38(4), 325 – 332.

[3] **A. Misevičius, A. Ostreika, A. Šimaitis, V. Žilevičius**. Improving local search for the traveling salesman problem. *Information Technology and Control*, 2007, *Vol*. 36(2), 187 – 195.

[4] **B. Yagmahan, M.M. Yenisey**. Ant colony optimization for multi-objective flow shop scheduling problem. *Computers & Industrial Engineering*, 2008, *Vol*. 54(3), 411 – 420.

[5] **C. Bohle, S. Maturana, J. Vera**. A robust optimization approach to wine grape harvesting scheduling. *European Journal of Operational Research*, 2010, *Vol*. 200(1), 245 – 252.

[6] **D. Pacciarelli, M. Pranzo**. Production scheduling in a steelmaking-continuous casting plant. *Computers and Chemical Engineering*, 2004, *Vol*. 28(12), 2823 – 2835.

[7] **D. Levišauskas, K. Jonelis, K. Brazauskas**. Adaptive Control of Temperature for Minimization Operating Costs of Industrial Methane Tank Process. *Information Technology and Control*, 2010, *Vol*. 39(3), 176 – 186.

[8] **D. Levišauskas, V. Galvanauskas, V. Čipinytė, S. Grigiškis**. Optimization of Feed-Rate In Fed-Batch Culture Enterobacter Aerogenes 17 E13 for Maximization of Biomass Productivity. *Information Technology and Control*, 2009, *Vol*. 38(2), 102 – 107.

[9] **E. Mokotoff, P. Chretienne**. A cutting plane algorithm for the unrelated parallel machine scheduling problem. *European Journal of Operational Research*, 2002, *Vol*. 141(3), 515–525.

[10] **F. Liu, G. Zeng**. Study of genetic algorithm with reinforcement learning to solve the TSP. *Expert Systems with Applications*, 2009, *Vol*. 36(3), 6995 – 7001.

[11] **G. Gutin, A. Yeo, A. Zverovich**. Traveling salesman should not be greedy: domination analysis of greedy-type heuristics for the TSP. *Discrete Applied Mathematics*, 2002, *Vol*. 117(1–3), 81 – 86.

[12] **J.A. De Loera, R. Hemmecke, S. Onn, R. Weismantel**. N-fold integer programming. *Discrete Optimization*, 2008, *Vol*. 5(2), 231 – 241.

[13] **J. Nenortaitė**. A Particle Swarm Optimization Approach in the Construction of Decision-Making Model. *Information Technology and Control*, 2007, *Vol*. 36(1A), 158 – 163.

[14] **K. Aardal, F. Eisenbrand**. Integer Programming, Lattices, and Results in Fixed Dimension. *Handbooks in Operations Research and Management Science*, 2005, *Vol*. 12, 171 – 243.

[15] **M. Li, Z. Yi, M. Zhu**. Solving TSP by using Lotka–Volterra neural networks. *Neurocomputing*, 2009, *Vol*. 72(16– 18), 3873 – 3880.

[16] **R. Tavakkoli-Moghaddam, A. Makui, S. Salahi, M. Bazzazi, F. Taheri**. An efficient algorithm for solving a new mathematical model for a quay crane scheduling problem in container ports. *Computers & Industrial Engineering*, 2009, *Vol*. 56(1), 241 – 248.

[17] **M.C. Georgiadis, A.A. Levis, P. Tsiakis, I. Sanidiotis, C.C. Pantelides, L.G. Papageorgiou**. Optimization-based scheduling: A discrete manufacturing case study. *Computers & Industrial Engineering*, 2005, *Vol*. 49(1), 118 – 145.

[18] **N. Boland, I. Dumitrescu, G. Froyland, A.M. Gleixner**. LP-based disaggregation approaches to solving the open pit mining production scheduling problem with block processing selectivity. *Computers & Operations Research*, 2009, *Vol*. 36(4), p.1064 – 1089.

[19] **N. Kliewer, T. Mellouli, L. Suhl**. A time–space network based exact optimization model for multi-depot bus scheduling. *European Journal of Operational Research*, 2006, *Vol*. 175(3), 1616 – 1627.

[20] **P.C. Chang, W.H. Huang, C.J. Ting**. Dynamic diversity control in genetic algorithm for mining unsearched solution space in TSP problems. *Expert Systems with Applications*, 2010, *Vol*. 37(3), 1863 – 1878.

[21] **S.L. Nonas, K.A. Olsen**. Optimal and heuristic solutions for a scheduling problem arising in a foundry. *Computers & Operations Research*, 2005, *Vol*. 32(9), 2351 – 2382.

[22] **T.P. Bagchi, J.N.D. Gupta, C. Sriskandarajah**. A review of TSP based approaches for flowshop scheduling. *European Journal of Operational Research*, 2006, *Vol*. 169(3), 816 – 854.

[23] **T. Bektas.** The multiple traveling salesman problem: an overview of formulations and solution procedures. *Omega*, 2006, *Vol.* 34(3), 209 – 219.

[24] **D. Dzemydiene, R. Dzindzalieta.** Development of architecture of embedded decision support systems for risk evaluation of transportation of dangerous goods. *Technological and Economic Development of Economy*, 2010, *Vol*. 16(4), 654–671.

[25] **D. Dzemydienė, A.A. Bielskis, A. Andziulis, D. Drungilas, G. Gricius.** Recognition of Human Emotions in Reasoning Algorithms of Wheelchair Type Robots. *Informatica*, 2010, *Vol*. 21, No. 4, 521–532.

[26] **D. Dzemydienė, R. Naujikienė, M. Kalinauskas, E. Jasiūnas.** Evaluation of security disturbance risks in electronic financial payment systems. *Intellectual Economics*, 2010, *No*. 2(8), 21-29.

[27] **G. Davulis, L. Šadzius.** Modelling and optimization of transportations costs. *Intellectual Economics*, 2010. *No*.1(7), 18-29.

Received December 2010.